



FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Título: Steer: Una aplicación móvil para el auto que mantiene a los conductores conectados e informados sin distraerlos de la tarea de conducir

Autores: Rodriguez Pedro - Villalba Joaquín

Director: Lic. Díaz Javier - Lic. Fava Laura

Codirector: --

Asesor profesional: Lic. Marra Juan

Carrera: Licenciatura en Sistemas - Licenciatura en Informática

Resumen

El concepto ciudades inteligentes es un término que abarca diversos tópicos, entre ellos, encontramos el de transporte y telecomunicaciones, que son fundamentalmente los temas tratados en la presente tesina. En este contexto, como búsqueda de la integración de la telefonía celular con los automóviles, surge Steer, una aplicación móvil de infoentretenimiento a bordo del automóvil. Para la elaboración de esta solución, se analizaron las soluciones actuales, se estudiaron diferentes tecnologías de desarrollo y comunicación y se implementó una aplicación móvil para el automóvil que se integra con el Sistema de Estacionamiento medido y alerta de eventos durante la conducción del vehículo.

Palabras Claves

Telemática. Smart Parking (Estacionamiento Inteligente). Smart Cars (Autos Inteligentes). Mirrorlink. Android. Teléfonos inteligentes. Sistemas de Infoentretenimiento (IVI System). Aplicación móvil. Aplicación web.

Conclusiones

Como conclusión, podemos decir que la tecnología utilizada, se adaptó correctamente para la solución propuesta, cumpliendo con los objetivos e ideas planteadas desde el principio.

Trabajos Realizados

Se diseñó e implementó una aplicación móvil, que permite conectar el sistema de infoentretenimiento del automóvil con el teléfono inteligente, logrando a partir de esto, una interacción fluida de la aplicación desde la posición del conductor del automóvil. El mismo, puede interactuar con el sistema de estacionamiento medido, alerta de eventos, búsqueda de caminos y seguimiento en tiempo real a otro automóvil.

Trabajos Futuros

Para trabajos futuros, nos resultaron interesantes los siguientes ítems:

- Automatización de creación de eventos para evitar hacerlos de manera manual.*
- Mejoras visuales de la aplicación.*

Universidad Nacional de La Plata

Facultad de Informática



STEER: Una aplicación móvil para el auto que mantiene a los conductores conectados e informados sin distraerlos de la tarea de conducir.

Tesina de Licenciatura en Informática y Licenciatura en Sistemas

Alumnos: Joaquín Villalba - Pedro Rodriguez

Directores: Lic. Javier Díaz - Lic. Laura Fava

Asesor Profesional: Lic. Juan Marra

Índice

Índice	1
Capítulo 1	3
Introducción	3
1.1 Motivación	3
1.2 Objetivos	4
1.3 Estructura de la tesis	5
Capítulo 2	7
Smart Cities, Smart Parking y Smart Cars	7
2.1 Smart Cities	7
2.2 Casos existentes	10
2.3 Smart Cars	12
2.4 Smart Parking	15
2.4.1 Historia	16
2.4.2 Desafíos de los sistemas propuestos de estacionamiento inteligente	16
2.4.3 Clasificación de los sistemas inteligentes de estacionamiento	17
2.4.4 Análisis y discusión de los resultados	19
2.4.5 Problemas y desafíos que enfrentan los sistemas de estacionamiento de hoy en día	19
2.4.6 Discusión de diferencias de investigación identificadas	20
2.5 Internet de las cosas	20
Capítulo 3	22
Estado del arte	22
3.1 Sistemas de infoentretenimiento	22
3.2 Sistemas para autos existentes	24
3.3 Tecnologías de desarrollo	26
3.3.1 Android Auto	26
3.3.2 CarPlay	30
3.3.3 Mirrorlink	35
3.4. Selección de la tecnología para el desarrollo de Steer	39
Capítulo 4	41
Propuesta de desarrollo	41
4.1 Justificación de la aplicación	41
4.2 MirrorLink	43
4.3 Herramientas utilizadas	46
Capítulo 5	55
Steer, el prototipo	55

5.1 Introducción	55
5.2 Diseño de la aplicación	55
5.2.1 Arquitectura del proyecto	55
5.2.2 Buenas prácticas y preferencia de herramientas	58
5.2.3 Material design y las buenas prácticas de diseño	59
5.2.4 Persistencia lado cliente	59
Capítulo 6	61
Steer, la aplicación	61
6.1 Aplicación móvil	61
6.2 Aplicación Web	74
Capítulo 7	83
Conclusiones y futuras líneas de investigación	83
7.1 Conclusión	83
7.2 Futuras líneas de investigación	84
Referencias Bibliográficas	86
Glosario	89

Capítulo 1

Introducción

1.1 Motivación

El aspecto y calidad de una urbe no sólo depende de la dotación de la misma respecto a infraestructuras físicas, servicios y entes institucionales, sino que en ellas incide cada vez más la disponibilidad y calidad de las comunicaciones, así como también la dotación en infraestructuras sociales. Los aspectos sociales e intelectuales son decisivos respecto a la competitividad urbana y a sus posibilidades de proporcionar una buena calidad de vida a los ciudadanos. Estas son algunas de las bases a partir de las cuales se ha construido el concepto de ciudades inteligentes o smart city, establecido como una herramienta para poder manipular y englobar características de un sitio urbano moderno y además, para poner en valor la importancia creciente de las Tecnologías de la Información y la Comunicación (TIC).

El continuo auge de las nuevas tecnologías nos lleva a encontrar una rama de las ciudades inteligentes llamada smart cars, que se centra en las tecnologías de información y entretenimiento a bordo (IVI). El infoentretenimiento es una palabra que reúne a las nuevas tecnologías que se vienen integrando a los automóviles, información y entretenimiento a bordo del mismo. Esta tecnología no ha surgido ahora, se remonta a la Fórmula 1, cuando a principios de los años 80 surgió la necesidad de saber que sucedía a bordo del automóvil en plena pista. Con esto, apareció la telemetría, concepto basado en sensores y transceptores de radiofrecuencia, que revolucionó la tecnología en este campo [1]. En años posteriores diferentes compañías de automóviles vieron que aparte de los sistemas de información a bordo dedicados a la seguridad de los automovilistas existían otras posibilidades. A partir de este concepto se han desarrollado aplicaciones diversas, desde algunas para mantener entretenido al conductor hasta otras para mantenerlo informado hacia y desde el exterior.

Por otro lado, los teléfonos móviles ofrecen una cantidad de aplicaciones de comunicación como Whatsapp, Facebook, Twitter, entre otras, que generan una constante recepción de notificaciones y mensajes de distinto tipo, lo que puede ocasionar distracciones cuando se maneja. La distracción al manejar es cualquier actividad que desvía la atención de la persona de la tarea principal que es conducir observando el contexto que

lo rodea. Todas las distracciones ponen en peligro al conductor, a los pasajeros y a los peatones.

El uso del teléfono representa la distracción más preocupante durante el manejo ya que requiere de la atención visual, manual y cognitiva del conductor. Según una encuesta realizada en 2015 un 30 % los conductores admiten usar el teléfono mientras conducen y un 75% dice haber visto a otros haciéndolo [2]. También es peligroso el uso de herramientas de navegación desde el teléfono si uno no se toma el trabajo de marcar los recorridos antes de comenzar con el manejo.

Con el propósito de reducir distracciones y los riesgos que conllevan, empresas automotrices y de tecnología han creado interfaces para poder conectar el celular con los dispositivos de navegación y entretenimiento del automóvil, con ellas, el conductor puede recibir notificaciones, y de ser necesario responderlas, pero desde una interfaz simplificada y utilizando los controles diseñados para este propósito, como lo son las pantallas táctiles, controles en el volante o comandos de voz. Muchas de estas interfaces son abiertas permitiendo a cualquier desarrollador crear distintas aplicaciones que, adoptando las normas de seguridad establecidas, mantengan al conductor conectado e informado sin interferir con la conducción.

En este contexto, tenemos por un lado teléfonos inteligentes que representan una gran herramienta ya que son de uso masivo y pueden ejecutar todo tipo de aplicaciones las cuales pueden proporcionar distintos datos del usuario ya sea mediante sensores, como la localización, o por acciones del usuario. Si bien son una distracción al momento de conducir actualmente existen las interfaces para poder comandar el dispositivo móvil a través de sistemas IVI del automóvil. Esto nos motivó, en conjunto con el CeSPI, a desarrollar una aplicación móvil que aproveche estos avances para adaptarla a los servicios brindados por este Centro de Cómputos de la Universidad, como el sistema de estacionamiento medido y las denuncias de eventos de la vía pública en tiempo real.

1.2 Objetivos

En esta investigación se busca exponer los aspectos más importantes del proceso de creación de Steer, una aplicación que busca aprovechar los grandes avances tecnológicos tanto en el área de los automóviles inteligentes como en la de los dispositivos móviles, ejemplos de ellos son el infoentretenimiento a bordo, por parte del automóvil, y la de los tipos de conexión de los dispositivos móviles, que han adquirido gran auge en los últimos años, como el 3G, 4G y WiFi. Son hechos que ayudan en gran medida a este

proyecto ya que sus bases se sientan en las nuevas tecnologías, nuevos servicios, seguridad en este tipo de campo, e investigación en un área que todavía no está explotada.

Es por ello que en Steer las interfaces de conexión entre celulares y automóviles sirven para ofrecer servicios o herramientas que faciliten la experiencia de manejo, y tengan un aporte respecto a este tipo de tecnologías, haciendo hincapié en la seguridad vial, más específicamente en la prevención de accidentes o atenuación de sus consecuencias.

Se pretende brindar distintos tipos de alertas y notificaciones con información pertinente al camino transitado, como por ejemplo accidente de tránsito, calles cerradas por obras, baches, salidas de colegios, etc. La aplicación debe actualizarse de manera constante y para esto se planea una retroalimentación a través de los mismos usuarios, facilitándoles la notificación de eventos a medida que circulan. A su vez los usuarios podrán emparejarse con otros e intercambiar información como rutas, puntos de interés o localización en tiempo real, entre otras.

Otra de los servicios que se pretende ofrecer es la integración con el sistema de estacionamiento medido de la ciudad de La Plata. Al estacionar en zonas medidas, los usuarios serán alertados y se les permitirá iniciar el estacionamiento así como la baja del mismo de manera manual o automática al volver a circular. Esto también nos permite, al realizar la baja de estacionamiento, alertar a otros usuarios potenciales de lugares libres. Es este otro servicio útil a la hora de transitar la ciudad.

Una de las principales premisas de la aplicación debe ser no distraer al conductor, por lo que cada acción debe demandar el menor tiempo posible, para esto se debe contemplar tanto el diseño visual (tipo y tamaño de letras, colores, tamaño de botones), como el funcional (priorizar alertas de voz, controles del volante, acciones predictivas).

Las contribuciones de esta tesina son:

- Una propuesta para nuevos ámbitos tecnológicos como son los smart cars.
- Un prototipo de aplicación implementado en algún SO móvil.
- Resultados de un experimento preliminar de la aplicación en tiempo real.

1.3 Estructura de la tesis

- Capítulo 1 – Introducción: Se plantea la motivación principal de la tesina, la problemática a la que hace referencia y los objetivos propuestos

- Capítulo 2 – Smart Cities, Smart Parking y Smart Cars: En este capítulo veremos una introducción a los conceptos de Smart Cities, Smart Cars y Smart Parking que representan el marco teórico.
- Capítulo 3 – Estado del arte: En esta sección desarrollamos el estado del arte de los Smart Cars en relación a los sistemas de infoentretenimiento, los tipos existentes y las tecnologías disponibles para el desarrollo de nuevas aplicaciones.
- Capítulo 4 – Propuesta de desarrollo: Justificamos el desarrollo propuesto y la tecnología elegida para el mismo. También se presentan las herramientas de software utilizadas para el desarrollo de la solución.
- Capítulo 5 – Steer el prototipo: En este capítulo vamos a explicar las decisiones tomadas en cuanto a la arquitectura y diseño de Steer, la aplicación a desarrollar.
- Capítulo 6 – Se hará un recorrido por cada una de las funcionalidades que fueron implementaron.
- Capítulo 7 – Conclusiones y Futuras líneas de investigación: Por último se expondrán las conclusiones generales tanto del proyecto como de la experiencia personal. A su vez, se presentarán posibles líneas de trabajo futuro.

Capítulo 2

Smart Cities, Smart Parking y Smart Cars

2.1 Smart Cities

La frase ciudades inteligentes no es nueva. Puede tener sus orígenes en el concepto de crecimiento inteligente, una teoría de planificación urbana y de transporte desarrollada a fines de los '90, que abogaba por nuevas políticas de planificación urbana. Dentro de sus objetivos está el de lograr un nuevo ámbito urbano en sentido de comunidad y de lugar de sus habitantes, y la expansión del rango de oportunidades de transporte, trabajo y vivienda, procurando preservar los recursos naturales y culturales.

La expresión ciudades inteligente ha sido adoptada desde el 2005 por una serie de empresas [Siemens, 2004][Cisco, 2005][IBM, 2009] para referenciar a la aplicación de sistemas de información complejos que integran la gestión de infraestructura urbana y servicios tales como edificios, transporte, distribución eléctrica, agua y seguridad pública. Desde entonces, ha evolucionado dando a entender casi cualquier forma de innovación basada en la tecnología de planificación, desarrollo y funcionamiento de las ciudades.

El uso intensivo de las TIC en las ciudades inteligentes puede producir varios beneficios:

- La reducción del consumo de recursos, particularmente el de energía y agua, ergo, puede contribuir a la reducción de emisiones de CO₂.
- Acrecenta la utilización de la capacidad de infraestructura existente, por lo tanto se intenta mejorar la calidad de vida y reducir la necesidad de proyectos de construcción tradicionales.
- Crear nuevos servicios y ponerlos a disposición de ciudadanos y viajeros, tal como, guías en tiempo real del funcionamiento de servicios de la ciudad.
- Los administradores de la ciudad pueden suavizar picos tales como lo son, la demanda de energía, agua y transporte que ocurren día a día, mediante el relevamiento de los mismos

Estos enfoques se han convertido en factibles gracias a los recientes avances tecnológicos:

- El uso de sensores y sistemas de control digitales para el control y funcionamiento de la infraestructura urbana: sensores de tráfico, sistemas de gestión de edificios, medidores de servicios digitales, entre muchos otros.
- La creciente penetración de las redes fijas e inalámbricas que permiten a este tipo de sensores y sistemas ser conectados a los centros de procesamiento distribuidos y, al mismo tiempo, el intercambio de información entre ellos.
- Desarrollo de nuevos algoritmos que permiten que los flujos de información sean analizados en parámetros cercanos al tiempo real, con el fin de proveer mejor rendimiento operativo.

Una característica común de estos sistemas es que los beneficios que ellos buscan conseguir se obtienen como resultado de los cambios en el comportamiento de los ciudadanos:

- Los conductores reciben mejor información sobre las condiciones del tráfico y de las rutas, para tomar mejores decisiones sobre el camino a seguir.
- Los consumidores reciben en tiempo real información acerca de su consumo de energía o acerca de los gastos monetarios que generan las mismas y tomar decisiones de consumo, lavadora, calefacción, refrigeración, carga de vehículos eléctricos son ejemplos de las mismas.
- Los servicios de emergencia comparten una vista en común ante una situación, lo que permite tomar una serie de decisiones de manera inmediata.

Entonces, hay aquí una toma de decisiones racionales que requieren información. Algunos participantes comprenderán el propósito del sistema y apoyarán sus metas u objetivos a través de sus propias tomas de decisiones. Otros participantes pueden, o bien no entender el sistema, o malinterpretar el mismo, y a partir de ello pueden tomar decisiones que no le sirven individualmente, ni contribuir a los beneficios colectivos. Otros también pueden entender el sistema y su funcionamiento pero oponerse a ello.

Así, aunque el enfoque de Smart Cities comienza como un intento pragmático, basado en la ingeniería, para mejorar el funcionamiento de la infraestructura y los servicios urbanos individuales, también podemos verlo como alterando inconscientemente las interacciones de los muchos sistemas dentro de una ciudad.[3]

Ante este escenario que muestra un entorno urbano con una demanda creciente de eficiencia, desarrollo sostenible, calidad de vida y gestión de los recursos, las administraciones públicas deben plantearse una evolución en los modelos de gestión de las

ciudades. Para ello, la aplicación de las tecnologías de la información y las comunicaciones (TIC) se hace imprescindible y se traduce en el concepto Smart City, que adelanta, con sus servicios, lo que ha dado en denominarse IoT concepto (que se refiere a la interconexión digital de objetos cotidianos a través de Internet, redes privadas u otros protocolos.) que definiremos más adelante. El concepto Smart City y el de Internet de las cosas son dos términos que van muy unidos. Ambos tienen en las comunicaciones M2M (máquina a máquina) su fundamento y adelantan, con sus aplicaciones y usos, la Internet del Futuro, que no solo consistirá en la conexión de cada vez más personas, sino en el planteamiento de un mundo digital en el que, idealmente, todo podrá estar conectado. Desde dispositivos, hasta objetos del mundo físico que habitualmente no disponían de esta conectividad; es el caso de los elementos urbanos, de los edificios, los coches, los electrodomésticos, los contadores, etc. y en general todo aquello que haya que gestionar o controlar.

La Smart City se convierte en una plataforma digital que permite maximizar la economía, la sociedad, el entorno y el bienestar de las ciudades, y facilita el cambio hacia un comportamiento más sostenible entre todos los agentes: usuarios, empresas y administración. Busca además aprovechar al máximo los presupuestos públicos, precisamente gracias a la mejora de los procesos propios de la ciudad y sus habitantes.

Por otro lado, permite habilitar nuevos modelos de negocio, constituyendo una excelente plataforma para la innovación en su entorno. En este contexto, una Smart City es un sistema complejo, un ecosistema en el que intervienen múltiples agentes, en el que coexisten muchos procesos íntimamente ligados y que resultan difíciles de abordar de forma individualizada.

De manera descriptiva, una Smart City es un espacio urbano con infraestructuras, redes y plataformas inteligentes, con millones de sensores y actuadores, dentro de los que hay que incluir también a las propias personas y a sus teléfonos móviles. Un espacio que es capaz de escuchar y de comprender lo que está pasando en la ciudad y ello permite tomar mejores decisiones y proporcionar la información y los servicios adecuados a sus habitantes. Además, el uso de técnicas analíticas avanzadas en tiempo real es lo que permite crear una especie de conciencia y entendimiento sobre la ciudad, lo que sin duda, mejora los servicios prestados.

En una Smart City la información adecuada llega en el momento preciso, integrando así “digitalmente” a las personas y a las cosas del entorno. Los espacios digital y físico se recombinan en la ciudad; por ello, la Smart City constituye un primer paso de la Internet de las cosas y por extensión, de la Internet del futuro tal y como se ha comentado.

Finalmente, hay que destacar que la característica smart de esta definición de ciudad no es duradera, es decir, no está asociada a la consecución de una meta en sí misma, sino que implica más bien el compromiso por parte de los distintos agentes involucrados en un proceso constante de mejora.

2.2 Casos existentes

A continuación vamos a repasar algunos trabajos que se relacionan con el concepto de smart cities y medios de transporte, tal como el tópico presentado en la presente tesina.

SmartSantanderRA

Se han situado sensores que miden el tráfico de entrada en la ciudad y más de 200 espiras que miden también la intensidad del tráfico. Además, se han instalado 10 paneles en otros tantos viales de la zona centro de la ciudad que informan al conductor en tiempo real y antes de que ingrese en el vial acerca de la disponibilidad de aparcamiento en superficie libre en dicha calle. Estos paneles proporcionan información de las casi 400 plazas de aparcamiento.

La ciudad cuenta también con etiquetas (tags) o pegatinas instaladas en las paradas de autobús, que facilitan datos acerca de las líneas, paradas, tiempos de espera, etc. de los autobuses municipales. Asimismo, dentro de la App. SmartSantanderRA, se dispone de información ampliada de las paradas de autobús que hay en el entorno, las líneas que paran en cada una, el tiempo que tardará en llegar el próximo autobús y la distancia exacta a la que se encuentra.

A esto se añade, en la misma aplicación, información sobre el tráfico, los puntos de alquiler de bicicletas, paradas de taxis o los aparcamientos subterráneos con datos en tiempo real sobre la disponibilidad en cada uno de ellos. Por lo tanto, se trata de una gran herramienta para la promoción de la movilidad multimodal, paradigma de la sostenibilidad urbana.

Por último, se han instalado cinco paneles informativos en puntos seleccionados de la ciudad con información en tiempo real sobre el estado del tráfico en distintas vías principales. Esto permite al usuario elegir entre distintas alternativas para llegar a su destino, con lo que reduce los tiempos de desplazamiento y contribuye activamente a la disminución de las emisiones de CO₂ (menos consumo de combustibles fósiles = menos emisiones de CO₂). También existe una aplicación para el pago del estacionamiento

limitado en superficie (Ordenanza Limitación Aparcamiento, OLA).

Todo ello ha generado algunos beneficios tanto para el funcionamiento del propio sistema como para la satisfacción de los ciudadanos. Aunque ya está generalizado el uso de GPS para el control de flotas, en lo que respecta al sistema de autobús municipal el avance para Santander radica en la mayor cantidad y calidad de la información que recibe el ciudadano, que conoce en tiempo real el tiempo que falta para que llegue su autobús, la distancia o la línea que debe tomar para llegar a un punto concreto (calidad de vida).[4]

Sistema Inteligente de Movilidad de Medellín

El incremento en la cantidad de vehículos, el crecimiento de la población, la urbanización, la industrialización y los cambios en la densidad de la población tienen un alto impacto sobre la movilidad de la ciudad, dado que traen como consecuencia la disminución de la eficiencia de la infraestructura de transporte y el incremento de los tiempos de viaje, de la contaminación del aire y del consumo de combustible.

Por todo lo anterior el Municipio de Medellín estudió la necesidad de implementar un SIMM desde 2010, que integra tecnologías de información, comunicaciones, la infraestructura de transporte y los diferentes tipos de vehículos.

Este sistema busca gestionar de forma eficiente todos estos componentes para mejorar la movilidad en la ciudad, mediante la optimización del uso de las vías y el mejoramiento de la seguridad, la disminución de los tiempos de desplazamiento, de la contaminación y del consumo de combustible, y la difusión de información al usuario para la toma de mejores decisiones de viaje.

El SIMM busca enfrentar los inconvenientes de movilidad a través de soluciones tecnológicas. Este es un caso exitoso de alianzas público-privadas y cuenta con siete componentes tecnológicos integrados en un Centro de Control de Tránsito para la vigilancia y gestión de la movilidad en la ciudad.

Para todo esto, el sistema utiliza los siguientes componentes:

- Cámaras de fotodetección
- Circuito cerrado de TV
- Paneles informativos
- Web
- Optimización de semáforos
- Software gestor

Este software recolecta, procesa y analiza datos del tráfico para la elaboración de estrategias y acciones que disminuyan los efectos negativos de la circulación vehicular, mejorando los tiempos de atención, regulación y control del tráfico y aportando a la disminución de la accidentalidad vial. Por tanto, es un sistema que toma las bondades del tránsito vehicular de la ciudad y a través de la tecnología, capta toda esta información y la gestiona de manera conveniente.[5]

2.3 Smart Cars

Uno de los puntos esenciales para aproximarnos a una ciudad inteligente es el control de la movilidad o tránsito en la misma y nos referiremos a cualquiera y todos los sistemas que mueven personas a través de una ciudad, como calles, vehículos, trenes, subtes, colectivos, bicicletas, etc, todos desempeñan un papel esencial en el ajetreo de las ciudades de hoy en día. Si pensamos en eficiencia, seguridad y costos, estos sistemas no siempre funcionan de la mejor manera. Las redes de transporte en ciudades de todo el mundo luchan con problemas graves, como la congestión.

En 2013, la congestión del tráfico robó la economía de EE.UU. \$ 124 millones de dólares, según un estudio de INRIX¹. Sin una acción significativa para aliviar la congestión, se espera que este costo para aumentar un 50% a \$ 186 mil millones para el año 2030. Otro estudio predice que las emisiones de los vehículos en marcha en los atascos de tráfico se traducirá en 1.600 muertes prematuras y \$ 13 mil millones en "costos sociales totales" en los EE.UU. en 2020. Por supuesto que no es un problema en los EE.UU. se enfrenta a solas. [6]

Un Índice 2014 de la congestión comparando los niveles de congestión en el 2014 en comparación con 2013, en más de 200 ciudades, clasificó las 10 ciudades más congestionadas:

1. Istanbul
2. Mexico City
3. Rio de Janeiro
4. Moscow
5. Salvadore
6. Recife
7. St. Petersburg
8. Bucharest

¹ Es una compañía global de Software que se especializa en servicios de automóviles conectados y análisis de transporte.

9. Warsaw

10. Los Angeles

Afortunadamente, hay un montón de maneras en que las ciudades puede solucionar la congestión del tráfico mediante la implementación de las TIC. La promesa de transporte inteligente y la realidad de la congestión de la ciudad hacen que este subsector o mercado este creciendo rápidamente. De acuerdo con Navigant Research, se espera que el mercado global de infraestructura y servicios de movilidad urbana inteligente crezca a de \$ 5.1 millones en 2015 a unos 25,1 \$ millones de dólares en 2024.

Bajo esta premisa surge el concepto de SmartCar, vehículos conectados que puedan acceder a la información provista por las ciudades inteligentes a través del internet de las cosas.

En la última década, los automóviles están evolucionando al punto que encontramos actualmente proyectos de vehículos totalmente autónomos lo que representa el máximo nivel de inteligencia.

Los niveles de automatización de un vehículo fueron clasificados por la SAE (Society of Automotive Engineers) en seis niveles que actualmente son tomados como el estándar en la materia. [7]

Nivel 0: Sin automatización

El coche no tiene ningún sistema automatizado que pueda tomar el control del vehículo, como mucho, puede tener sistemas que emitan alguna advertencia.

Nivel 1: Asistencia del conductor

A este nivel pertenecen los coches que comenzaron a incluir sistemas de control de crucero adaptativo o tecnología para mantener al coche en el carril.

Nivel 2: Automatización parcial

En este nivel el coche comienza a poder denominarse semiautónomo. El conductor está obligado a estar alerta para tomar el control en cualquier momento ya que el coche puede no responder adecuadamente. Es obligatorio que el sistema se desactive cuando el conductor toma el control.

Nivel 3: Automatización condicional

Los coches pueden circular de forma autónoma en entornos controlados como autopistas.

Nivel 4: Alta automatización

Los coches autónomos pueden circular sin supervisión del conductor en áreas restringidas donde el coche tenga suficiente información para no depender del conductor.

Nivel 5: Automatización completa

La conducción autónoma en este nivel es completa. El coche puede circular por cualquier carretera o ciudad siempre y cuando sea legal. La tecnología permitiría al coche poder reaccionar ante cualquier imprevisto.

Desde hace años hay proyectos de vehículos totalmente autónomos. Tenemos el caso de Google que desde 2009 decidió sumarse a la investigación de esta área, dándole un nuevo impulso. La compañía comenzó modificando modelos existentes, de marcas como Toyota, Audi o Lexus, a los que incorporó múltiples cámaras, sensores y software de control, pero fue a finales del pasado 2014 cuando presentó su propio prototipo. Los coches de Google carecen de volante, acelerador o freno. Solo tienen un botón de arranque y uno de parada. Está repleto de sensores que eliminan los puntos ciegos y puede detectar objetos en todas direcciones a una gran distancia; su velocidad está limitada a 40 km/h y se controla mediante una aplicación móvil en la que el usuario elige su destino.

El estado de California (EEUU) dio permiso a ocho marcas de coches a realizar pruebas de vehículos sin conductor en carreteras públicas en septiembre de 2014. Desde entonces, el Departamento Californiano de Vehículos de Motor ha reportado seis accidentes de este tipo de coches y en cinco de ellos se vio involucrado un modelo de Google, aunque la culpa fue del conductor del otro automóvil. Desde que la compañía iniciara su andadura en la conducción autónoma, ha reconocido 11 incidentes, todos menores.

Otro ejemplo es el de la compañía Tesla que en octubre de 2016 ha anunciado todos sus vehículos de Tesla producidos en su fábrica, desde el Modelo 3, contarán con el hardware necesario la conducción autónoma a un nivel de seguridad sustancialmente mayor que el de un conductor humano. Sus modelos contarán con ocho cámaras envolventes que proporcionan una visibilidad de 360 grados alrededor del automóvil a una distancia de hasta 250 metros. Doce sensores ultrasónicos complementan esta visión, permitiendo la detección de objetos duros y blandos a casi dos veces la distancia del

sistema anterior. Un radar orientado hacia adelante con procesamiento mejorado proporciona datos adicionales sobre el mundo en una longitud de onda redundante, capaz de ver a través de fuertes lluvias, niebla, polvo e incluso el automóvil que se encuentra delante. Su computadora a bordo ejecutará la nueva red neuronal desarrollada por Tesla para software de procesamiento de visión, sonar y radar. Este sistema proporciona una vista del mundo a la que un conductor no puede acceder, viendo en todas las direcciones simultáneamente y en longitudes de onda que van más allá de los sentidos humanos. [8]

Este tipo de proyectos son una realidad hoy en día pero aún no se han impuesto en el mercado global, sólo en ambientes permitidos y medianamente controlados. Lo que sí encontramos en todas las carreteras son los vehículos conectados. Desde hace unos años, los fabricantes de automóviles y desarrolladores de tecnología han creado sistemas que dotan a los vehículos de conectividad, y en los últimos años se ha multiplicado el número de automóviles que lo van a incorporar de fábrica. Según las previsiones de la consultora especializada en automoción SBD y el GSMA (Groupe Speciale Mobile Association), el mercado de vehículos conectados generará en 2018, 39 mil millones de euros frente a los 13 mil millones de 2012.

El objetivo de estos sistemas es permitir al conductor manejar su móvil desde el coche (hacer llamadas, consultar mapas, escuchar sus correos) pero también le proporcionan información sobre el estado del vehículo y alertas de posicionamiento, si se tiene activado el GPS. Pueden incluso, avisar a los servicios de emergencia de un accidente.

Normalmente, todas esas funcionalidades se gestionan desde el panel de control integrado en el tablero, y ya estudian otras opciones como la realidad aumentada para convertir el parabrisas en la pantalla sobre la que se proyecta toda esta información, y evitar así, un punto de distracción para el conductor. Estos sistemas infoentretenimiento serán analizados más adelante.

2.4 Smart Parking

Nos encontramos aquí con dos grandes problemáticas. La primera es el número de publicaciones escritas en este dominio, que es limitado. El segundo problema está relacionado con esta prometedora tecnología (estacionamiento inteligente); éste aún no se encuentra de manera extendida en el mundo real para la mayoría de los países, especialmente los países en desarrollo, donde las personas tienen más necesidad de aliviar el estrés y reducir la cantidad de tiempo dedicado a la búsqueda de espacios para

estacionar. Tales mejoras conducirán a una disminución en la contaminación y muchos otros problemas que se mencionan en el artículo.

2.4.1 Historia

Parece que la palabra "inteligencia" o "inteligente" tiene diferentes significados de acuerdo con los requisitos de las personas y su ubicación en el paso del tiempo. A través de los años, ha aumentado la cantidad de tecnologías relacionadas con el sistema de estacionamiento, éstos han existido casi desde el momento en que se inventaron los automóviles. En cualquier época donde ha habido una cantidad significativa de tráfico, ha habido sistemas de estacionamiento de automóviles. Los sistemas de estacionamiento de automóviles se desarrollaron por primera vez a principios del siglo XX en respuesta a la necesidad de espacio para aparcar los vehículos. De hecho, el uso de sistemas de estacionamiento e-smart se remonta a la década de 1920, cuando aparecieron sistemas de estacionamiento automáticos en ciudades de Estados Unidos como Los Ángeles, Chicago, Nueva York y Cincinnati.

A mediados de los 80, los sistemas utilizados para estacionar se basaban principalmente en el método tradicional de presionar un botón en el dispositivo junto al punto de control para obtener un ticket de estacionamiento y al salir, el conductor debe pagar antes de insertar su ticket para que la barrera se eleve. Este era el método utilizado para determinar cuántos automóviles ingresaban y salían del sistema cada día, y se utilizó para contar la cantidad de espacios libres disponibles. Se comenzó utilizando diferentes métodos, como sensores o barreras, para poder conocer el estado de los estacionamientos.

2.4.2 Desafíos de los sistemas propuestos de estacionamiento inteligente

Hoy en día, los principales sistemas de transporte día tienen que enfrentarse a grandes desafíos en relación con los sistemas de estacionamiento especiales, para los cuales los ingenieros y diseñadores de ciudades inteligentes deben estar preparados. Numerosos estudios recientes han llevado a la conclusión de que se necesitan nuevos sistemas inteligentes de estacionamiento, en casi todas las ciudades metropolitanas del mundo, especialmente en los próximos años, para aliviar muchos problemas, tales como el consumo de nafta y la emisión de contaminantes, para ahorrar tiempo y reducir frustración al momento de buscar un espacio de estacionamiento.

Por lo tanto, para que cualquier sistema propuesto se considere inteligente en relación con el proceso de estacionamiento, debe tener como mínimo, los siguientes factores y especificaciones: Brindar orientación a los usuarios sobre el estacionamiento disponible.

- Ser capaz de detectar con precisión la ocupación del vehículo en tiempo real.

- Brindar orientación a los usuarios sobre el estacionamiento disponible.
 - Que simplifique la experiencia de estacionamiento y agregar información a los interesados en el sistema de estacionamiento, tal como el caso de los conductores.
 - Que permita que la toma de decisiones inteligentes utilizando datos, incluyendo aplicaciones de estado en tiempo real e informes analíticos históricos.
 - Ser capaz de proporcionar al usuario toda la información necesaria sobre el estado de cualquier cambio en el área de estacionamiento que pueda ocurrir en tiempo real.
- Estos desafíos deben abordarse desde el principio para garantizar que el sistema funcione de manera eficiente [9]

2.4.3 Clasificación de los sistemas inteligentes de estacionamiento

Sistemas de estacionamiento inteligentes basados en el modelo de agente

Estos tipos de sistema pueden ser cualquier entidad capaz de observar hechos a través de sensores, ya que el sistema actúa sobre los cambios del entorno a través del intercambio de información e interacción sobre esos hechos. Tiene características útiles, como la autonomía, reactividad y adaptabilidad (Mahmud 2013). Esencialmente, un sistema multiagente es un método de modelado desarrollado para representar sistemas con entidades, autonomía e interacción.

Sistemas basados en redes de sensores inalámbricos

Son la técnica más popular en la última década entre los investigadores, ya que las redes de sensores inalámbricos tienen varias ventajas, como flexibilidad, inteligencia, costo razonable, despliegue rápido, y por lo general, consta de nodos sensores. Este sistema permite que un automóvil detecte la entrada al estacionamiento y guíe al conductor de manera eficiente hacia un espacio de estacionamiento vacío mediante carteles que se le muestran al conductor. Para cada estacionamiento individual, se fija un sensor en el techo sobre cada uno de ellos, y los sensores ultrasónicos funcionan en función de la ecolocación. El sensor transmite un sonido que golpea un objeto sólido (automóvil o tierra) y se refleja de vuelta al sensor. En la solución centralizada, algunos automóviles también están equipados con sensores ultrasónicos, que pasan más allá de los espacios de estacionamiento para recopilar datos de ocupación y cargar los datos a la base de datos centralizada. Los autos que necesitan estacionarse simplemente consultan la base de datos centralizada. (Lee et al., 2008) propusieron el uso de una combinación de sensores magnéticos y ultrasónicos para la detección precisa y confiable de vehículos en un estacionamiento.

Sistemas de estacionamiento inteligentes basados en sistemas de posicionamiento global o GPS

La tecnología de Sistemas de Posicionamiento Global se usa para determinar y rastrear la ubicación precisa de un vehículo. Se utiliza para ofrecer información sobre la ubicación y la disponibilidad de espacios de estacionamiento. Propuesto por (Pullola, 2007; Chon 2002, Hanif, 2010), presentaron un sistema basado en la ubicación, llamado NAPA. El servidor en el sistema asocia edificios en el área con estacionamientos en el orden de las distancias a un edificio. Después de localizar el estacionamiento disponible más cercano, el usuario envía un mensaje al servidor de NAPA que ha estacionado. Luego el servidor actualiza la información sobre el lote en consecuencia. Cuando el usuario abandona el estacionamiento, el servidor de NAPA puede cobrar automáticamente la tarifa adecuada si es necesario. (Hanif, 2010) propuso un nuevo sistema de estacionamiento inteligente que utiliza servicios de SMS. Este sistema es capaz de encontrar espacios de estacionamiento en áreas específicas de estacionamiento de automóviles. Un sistema de reserva de estacionamiento se desarrolla de tal manera que los usuarios pueden reservar sus lugares de estacionamiento a través de servicios de SMS usando el GPS.

Sistemas de estacionamiento inteligentes basados en visión por computadora

Este campo de estudio incluye métodos para adquirir, procesar y analizar imágenes. Utiliza computadoras para emular la visión humana, incluido el aprendizaje y la capacidad de hacer inferencias y tomar medidas basadas en las entradas visuales, conocidas como visión por computadora. El objetivo de la visión por computadora es hacer que las computadoras perciban y procesen de manera eficiente los datos visuales, como imágenes y videos, y actúen según los cambios en estas imágenes. Por lo general, la técnica implica analizar unos cuadros por segundo y luego enviar los datos a una base de datos central, después de lo cual, el usuario puede recuperar información sobre los cambios en el estacionamiento. En (Takizawa, 2004), su sistema utilizó CCTV en un flujo de detección de vehículos para detectar la presencia de un automóvil o vehículo en un estacionamiento particular. La detección de píxeles se usa para detectar la presencia de un vehículo en cada aparcamiento. Se utiliza un cierto número de píxeles en la escala de grises como umbral para diferenciar los píxeles del vehículo y del lote desocupado. Este es un ejemplo de los tantos existentes en la actualidad.

Sistemas de estacionamiento inteligentes basados en identificación por Radiofrecuencia o RFID

Las soluciones de RFID para estacionamiento inteligente hacen posible administrar el estacionamiento fácilmente, especialmente en las etapas de prototipo. El mecanismo principal de la tecnología RFID depende de un campo electromagnético para identificar y rastrear automáticamente las etiquetas adheridas a los objetos. Su sistema usa un software para controlar e informar los cambios en el estado del espacio de estacionamiento, y para la operación de tareas tales como elegir el espacio libre más cercano, y luego envía esta información al conductor.

Otros sistemas híbridos, M2M, IoT

Se propuso un sistema inteligente que detecta y encuentra la ubicación para estacionar a través de sistemas basados en IoT, sensores inteligentes y actuadores, con el middleware que conecta los clientes con dispositivos terminales. El sistema depende de la comunicación Bluetooth entre el teléfono inteligente y las motas inalámbricas del sensor.[10]

2.4.4 Análisis y discusión de los resultados

En relación con los puntos anteriores, nos encontramos con varias técnicas disponibles, especialmente tecnología de estacionamiento inteligente basada en sensores. Numerosos sistemas de estacionamiento inteligente han sido elegidos para cumplir el mismo propósito con una técnica diferente y los datos han sido analizados de acuerdo a diferentes factores, tales como costo de la técnica, confiabilidad, escalabilidad, precisión, tipo de comunicación, complejidad del circuito, método de operación y facilidad de instalación.

2.4.5 Problemas y desafíos que enfrentan los sistemas de estacionamiento de hoy en día

A medida que la sociedad crece, cada vez hay más sistemas basados en computadoras para mejorar la eficiencia de los mismos, los sistemas de estacionamiento inteligente son cada vez más numerosos. Los sistemas inteligentes de estacionamiento se han convertido en una necesidad, especialmente, en las grandes urbes. Al diseñar con un enfoque centrado en el usuario, surgen dos problemas principales; en primer lugar, los problemas operativos y, en segundo lugar, los desafíos técnicos específicos. Estos desafíos deben abordarse desde el principio para garantizar que el sistema funcione de manera eficiente in situ. Todos los estudios disponibles relacionados con los sistemas de estacionamiento inteligente tradicionales en la última década, indican que no satisfacen los requisitos de los conductores.

2.4.6 Discusión de diferencias de investigación identificadas

En resumen, cada solución tiene diferentes requisitos tecnológicos en relación con todos los factores mencionados anteriormente, utilizando diferentes métodos para facilitar los servicios de estacionamiento para el usuario. Cada prototipo del sistema tiene ventajas y desventajas en términos de los siguientes criterios: costo, confiabilidad, escalabilidad, precisión, tipo de comunicación, complejidad del circuito, confiabilidad, método de operación y facilidad de instalación / usabilidad del sistema. Muchos tipos de investigación han influido en la forma en que se han diseñado los sistemas existentes, midiendo su desempeño y calidad desde la perspectiva del diseñador, ingeniero o desarrollador.

2.5 Internet de las cosas

Por lo general, el término Internet de las Cosas (IoT – Internet of things) se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. Sin embargo, no existe ninguna definición única y universal.

Al igual que con varios conceptos novedosos, las raíces de IoT se pueden remontar al Instituto de Tecnología de Massachusetts (MIT), hasta llegar al trabajo del Auto-ID Center.

Este grupo, fundado en 1999, realizaba investigaciones en el campo de la identificación por radiofrecuencia en red (RFID) y las tecnologías de sensores emergentes. Los laboratorios de investigación estaban conformados por siete universidades ubicadas en cuatro continentes, seleccionadas por Auto-ID Center para diseñar la arquitectura de IoT.

En 2003, había aproximadamente 6,3 mil millones de personas en el planeta, y había 500 millones de dispositivos conectados a Internet. Si dividimos la cantidad de dispositivos conectados por la población mundial, el resultado indica que había menos de un dispositivo (0,08) por persona. De acuerdo con la definición de Cisco IBSG, IoT aún no existía en 2003 porque la cantidad de cosas conectadas era relativamente escasa.

Actualmente, IoT está compuesta por una colección dispersa de redes diferentes y con distintos fines. Por ejemplo, los automóviles actuales tienen múltiples redes para controlar el funcionamiento del motor, las medidas de seguridad, los sistemas de comunicación y así sucesivamente. De forma similar, los edificios comerciales y

residenciales tienen distintos sistemas de control para la calefacción, la ventilación y el aire acondicionado, la telefonía, la seguridad y la iluminación. A medida que IoT evoluciona, estas redes y muchas otras estarán conectadas con la incorporación de capacidades de seguridad, análisis y administración.[11]

Capítulo 3

Estado del arte

3.1 Sistemas de infoentretenimiento

La información que podemos obtener del uso de los smartphones puede ser muy útil para planificar y administrar el tránsito en una ciudad. Muchos de estos datos se pueden obtener si el usuario está efectivamente usando el teléfono celular al momento de conducir, pero como expusimos en la introducción de esta tesis, esto presenta un problema que actualmente preocupa por la cantidad de accidentes que se reportan.

Entonces identificamos algunos problemas o desafíos ¿cómo debería ser posible hacer efectivamente la comunicación entre un vehículo y un teléfono? ¿Cómo evitar la distracción del conductor durante la interacción con la aplicación? ¿Cómo asegurar que la solución elegida permanecerá viable a través de la vida útil del vehículo?

Un sistema IVI, es un conjunto de hardware y software en el automóvil que provee entretenimiento de audio y/o video. Los sistemas IVI originó que los sistemas de audio automotriz que consistían de radios, cassette o CD players, ahora incluyan sistemas de navegación automotriz, reproductores de video, conectividad USB y Bluetooth, carputer, internet vehicular y WiFi. Una vez controlado por el tablero del automóvil, los sistemas IVI pueden incluir manipulación a través del volante y comandos de voz del vehículo.

El marco para la toma de decisiones consistiría, en primer lugar, en la ubicación de la aplicación. Podría ser almacenado en la unidad de infoentretenimiento (en otras palabras en el sistema IVI), o en el teléfono. En segundo lugar, está la cuestión de cómo controlar la distracción del conductor; qué interfaz se debe utilizar, qué aplicaciones se deben permitir, y la forma de proceder con la legislación y la certificación. La tercera pregunta sería acerca de la estrategia de integración.

Actualmente hay una gran cantidad de opciones y tecnologías diferentes, esto se debe, simplemente, al hecho de que el concepto es nuevo y emergente. Ahora las empresas están promoviendo soluciones que son relevantes para sus estrategias, las métricas de negocio y otros hechos que los estimulan para forzar una opción sobre otra. Lo bueno es que algunas de las tecnologías pueden coexistir a bordo de un coche, pero el riesgo es que podría generar cierta confusión para los consumidores, lo que va de la mano con la idea de evitar distracciones. También es natural suponer que, en algún momento,

habrá sólo dos o tres tecnologías de integración ganadoras de la corriente principal, y el resto permanecer, posiblemente, como tecnologías especializados y de nicho.

Lo que ha ocurrido en los últimos años es que cada fabricante de automóviles, así como cada vendedor de teléfonos ha establecido un enfoque muy sistemático para este dominio, y ahora es el momento de avanzar hacia aplicaciones y servicios reales en lugar de perder tiempo en la tecnología de integración. Nuevas soluciones, obviamente, deben planificarse con la consideración de la arquitectura modular, lo que permite migrar fácilmente de una tecnología de integración a otra, al menor costo posible y con la menor influencia en la experiencia del usuario.

La primer decisión que planteamos fue definir el lugar donde se ejecutará la aplicación. Tenemos dos alternativas.

Incorporación de aplicaciones dentro del IVI

La solución obvia es la de crear un sistema operativo propietario del IVI. Pero entonces el fabricante de automóviles necesita desarrollar sus propias aplicaciones y/o adaptar las existentes en su propio sistema. Generalmente esto no está relacionado directamente con los intereses interés de las automotrices. El mecanismo de distribución sería, también, potencialmente problema. La industria móvil ha logrado establecer los ecosistemas alrededor de cada sistema operativo móvil, como el último paso en la evolución de la distribución de aplicaciones. En el caso de las automotrices, este ecosistema, no existe en la actualidad y la creación de uno nuevo sería muy difícil, aun mas teniendo en cuenta la cantidad de automotrices. Otro problema que podría impedir la creación de sistemas operativos y software independientes sería el mantenimiento del hardware capaz de ejecutar las aplicaciones más nuevas cada año; los ritmos de investigación y desarrollo entre las industrias de automotriz y electrónica son muy distintos.

A pesar de estos problemas y retos, esta es la solución más común para los IVI en el mercado asiático, en este momento. Esto se logra adaptando el sistema operativo Android. Hay algunos inconvenientes en esta opción. La primera es que este sistema es demasiado inmaduro para este uso, los componentes de este sistema fueron diseñados para manejar teléfonos o tabletas, en lugar de las IVI (pero esto podría potencialmente cambiar con el tiempo). En segundo lugar, la distracción del conductor no se maneja de forma alguna. En esta solución, no hay integración clara, el IVI es sólo la reutilización de una aplicación que ya existe en el mercado.

La alianza GENIVI, un consorcio sin fines de lucro, fue fundada para abordar algunas de estas cuestiones. Se ha comprometido a crear una plataforma de desarrollo de código abierto para los IVI. Fabricantes como BMW, PSA, Renault, Nissan, GM y Jaguar

Land Rover están trabajando juntos para establecer un ecosistema [12]. Comarch es también parte de esta alianza como proveedor del middleware y servicios. GENIVI puede ganar algo de impulso a efecto de la escala y principios muy razonables. El problema principal con GENIVI es simplemente el tiempo. Los ecosistemas móviles están tan bien establecidas para los consumidores, que una nueva categoría de aplicaciones (para aplicaciones de automóviles) puede alcanzar un alto potencial y asentarse mucho más rápido que un nuevo ecosistema basado en GENIVI. [13]

Aplicaciones embebidas en el teléfono

Como se ha mencionado antes, la comunicación y reproducción de música desde un teléfono es imposible sin usar las manos, pero, con las interfaces correctas y a través de los modernos enlaces entre un teléfono y un sistema IVI, otorgar estas y otras funcionalidad de manera confortable e intuitiva.

La idea de llevar las aplicaciones del teléfono al automóvil es muy interesante para muchas personas si logra con eso utilizar cómodamente a la pantalla de panel de control, a los altavoces, micrófono que han sido específicamente diseñado y preparados para el interior de su auto.

Mantener las aplicaciones embebidas en el teléfono e integradas al vehículo crea muchas oportunidades, siempre que sean bien diseñados y contruidos por organizaciones que son conscientes de las necesidades específicas en lo referido a automóviles. El elemento crucial de una aplicación exitosa es la interfaz hombre-máquina bien diseñada y la experiencia de usuario en general, mantener el equilibrio adecuado de las limitaciones del coche, y las experiencias obtenidas directamente de las plataformas móviles modernas.

3.2 Sistemas para autos existentes

Sync AppLink es el proyecto de vehículo conectado de Ford. Utiliza el sistema de conexión a Internet a través de la conexión Bluetooth de un smartphone. De esta forma, el coche se conecta al dispositivo y utiliza el plan de datos ya contratado en ese dispositivo. El sistema depende del teléfono para funcionar, y sin éste, Ford Sync no es operativo.

Entre sus funcionalidades, el conductor de un vehículo equipado con Ford Sync no sólo puede hacer que el coche lea un SMS que acaba de recibir en su teléfono, sino que puede pedirle información sobre otros temas, como el tiempo, compartir información de aplicaciones o consultar los horarios del cine más cercano, todo ello a través de un sistema de reconocimiento de comandos por voz.

Los desarrolladores pueden adaptar sus aplicaciones al funcionamiento de AppLink a través del SDK proporcionado por Ford, permitiendo que puedan ser controladas mediante el asistente vocal del coche, algunas teclas del volante y los mandos de la consola central. En el caso de que el usuario tenga un iPhone, deberá conectar el teléfono al coche mediante un cable USB. Dentro de la página del programa de desarrolladores se establecen una serie de criterios estrictos para la inclusión de una aplicación en los vehículos dotados de SYNC AppLink. Ciertas categorías de aplicaciones están directamente prohibidas, como pueden ser:

- Aplicaciones de vídeo.
- Aplicaciones que muestren mucha cantidad de texto en la pantalla.
- Aplicaciones con mucho contenido de imágenes o vídeos.
- Juegos de cualquier tipo.

Estos estrictos criterios de seguridad están basados en las directrices de la National Highway Traffic Safety Administration (NHTSA).

Caso distinto es el Citroën y su sistema Multicity Connect, si bien en principio la interfaz presentada es similar a la de Sync AppLink ya que cuenta con una pantalla táctil y controles al volante, la gran diferencia en este caso es la imposibilidad de integrar aplicaciones del smartphone en la pantalla del coche. Solo se puede tener acceso a las aplicaciones proporcionadas por la marca. Con lo cual no se puede realizar aplicaciones para este sistema.

Otros fabricantes como BMW y AUDI adoptan los dos modelos, sus sistemas poseen aplicaciones embebidas directamente en el sistema IVI, pero además puede conectarse el teléfono móvil, y ejecutar aplicaciones directamente desde el mismo, para controlarlo desde el dispositivo de infoentretenimiento. Para las aplicaciones embebidas apuestan por integrar directamente una tarjeta SIM en sus automóviles, ofreciendo independencia del teléfono de su conductor. Los coches pasan a tener su propia conexión particular, lo que puede ser una desventaja ya que es necesario pagar una línea adicional.

En el caso de BMW y su sistema ConnectedDrive solo se pueden adaptar aplicaciones de sistema operativo iOS. Audi, su sistema se llama Audi Connect, posee aplicaciones para Android y iOS, pero estas son desarrolladas por la propia marca.

De esta forma podemos seguir nombrando sistemas similares desarrollados por otros fabricantes de automóviles al punto que podemos encontrar un sistema para cada marca. Esta fragmentación de sistemas de conectividad para integrar el teléfono en los coches es el problema predominante en el panorama automovilístico actual. La creación de

un estándar de conectividad único que supere esta fragmentación ha llevado al desarrollo de tecnologías como MirrorLink, CarPlay, Android Auto.

Muchas marcas han integrado estas tecnologías a sus sistemas, incluso las que ya poseían algún tipo de interfaz, como las que mencionamos antes, porque esto le permite incorporar aplicaciones que pueden ser definitorias para el usuario a la hora de decidirse por un modelo u otro. También es un punto favorable para los desarrolladores ya que pueden adoptar sus aplicaciones a muchos modelos de vehículos en vez de tener que adaptarse a muchas interfaces, una por cada marca y en algunos casos según el modelo.

[14]

3.3 Tecnologías de desarrollo

Con lo descrito anteriormente hemos reducido la selección de la tecnología de desarrollo en las tres grandes plataformas que encontramos actualmente. Serán las que analizaremos en detalle:

- Android Auto
- CarPlay
- Mirrorlink

3.3.1 Android Auto

Es un estándar de telemática desarrollado por Google para permitir que los dispositivos móviles que corren sobre el sistema operativo Android (Versión a partir de Lollipop) puedan operar el panel de control principal del automóvil. Esto proveerá el control de funciones tal como GPS, mapa/navegación, reproducción de música, mensajería SMS, telefonía, búsqueda en la web, de manera segura durante un viaje.

Respaldada por muchos fabricantes de autos, entre ellas Seat, y promovida por la Open Automotive Alliance (OAA), un grupo abierto de tecnología y compañía de automoción que se unen con el fin de lograr maximizar las ventajas del SO Android y al mismo tiempo ayudar a minimizar la distracción al usuario.

El objetivo de esta plataforma es extender la funcionalidad de un dispositivo móvil que use el SO Android en el vehículo usando el sistema IVI. Esto se logrará mediante la conexión del celular vía USB con sistema de infoentretenimiento del automóvil.

Interfaz

La interfaz de Android Auto mantiene el mismo diseño conocido de Android y de diferentes servicios de Google, pero los simplifica para ofrecer texto más grande y una interfaz un poco más gráfica. El objetivo principal de esto es lograr que el conductor no se distraiga al conducir.

Android Auto podrá controlarse desde la pantalla del tablero táctil, desde los controles del volante u otros controles integrados en el vehículo.

Asimismo, Google integra su reconocimiento de voz para permitir que el usuario no tenga siempre que interactuar con el sistema mediante el uso de botones. De manera que se puede realizar múltiples tareas mediante el uso de la voz.

Arquitectura

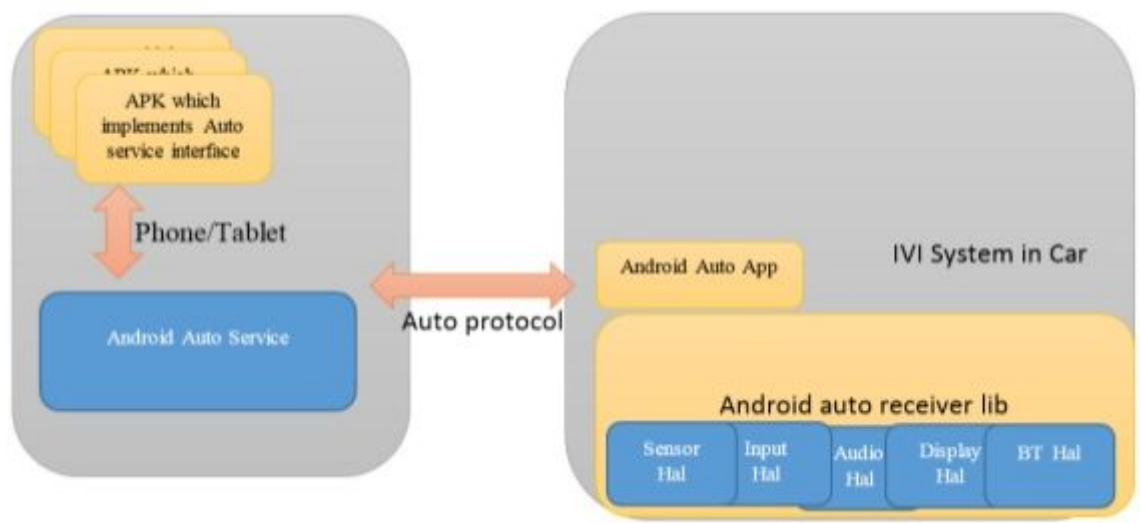


Figura 3.1. La aplicación Android y los servicios están corriendo en el dispositivo móvil.

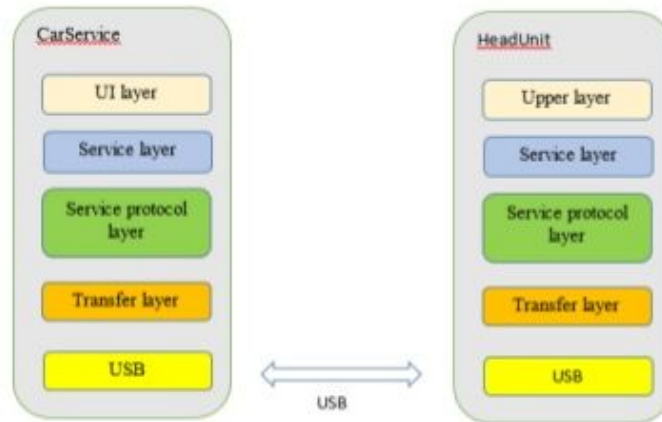


Figura 3.2. La implementación de Car Service y HeadUnit puede ser dividida en cinco capas.

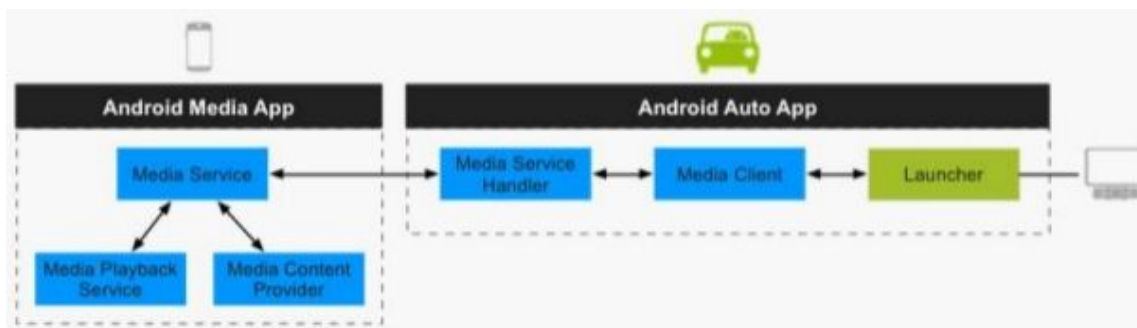


Figura 3.3. La aplicación de Android Auto funciona como un proxy en el automóvil.

Manejo seguro

Las aplicaciones que funcionan con una interfaz de usuario deben minimizar las distracciones a las que se enfrenta el conductor, a través de mejores prácticas, como lo son el uso de comandos de voz y una interfaz simple y sencilla de usar. Es lo que se pretende cuando se diseña una aplicación para Android Auto.

Es importante que la aplicación cumpla con una serie de criterios establecidos por Android antes de que la misma pueda estar disponible para su uso. Existen dos maneras de probar la aplicación:

- Usando un simulador de Android Auto para validar cada punto de la lista que propone Android.
- Probar la aplicación en un vehículo que posea soporte la tecnología necesaria..

Limitaciones

- La manera de poder crear la comunicación desde el dispositivo móvil al sistema IVI del automóvil es solamente vía cable USB, dejando de lado muchos otros tipos de conectividad que tenemos en la actualidad, por ejemplo, NFC, Bluetooth, WiFi.
- La interfaz que nos ofrece es muy limitada. Este quizá es uno de los puntos más débiles de Android Auto. Para el desarrollo de una aplicación sólo se nos permite elegir entre dos tipos de desarrollo. Ellas son, aplicaciones de audio y aplicaciones de Mensajería. Con la primera de éstas tendremos disponible un reproductor de audios, puede ser por ejemplo un audiolibro o un reproductor de música. Con el segundo de ellos se puede recibir mensajes o enviar un mensaje escrito por defecto, configurable desde la aplicación. Hasta el momento no se puede hacer una aplicación genérica segura de Android sobre este simulador.

Ventajas

- Android es el Sistema Operativo más utilizado en el mundo, por lo cual posee la gran ventaja de altísima cantidad de dispositivos que tienen de los consumidores. La *Figura 3.4* muestra estadísticas de ventas a usuarios finales, de teléfonos inteligentes y los Sistemas Operativos de los mismos.

Worldwide Smartphone Sales to End Users by Operating System in 2017 (Thousands of Units)

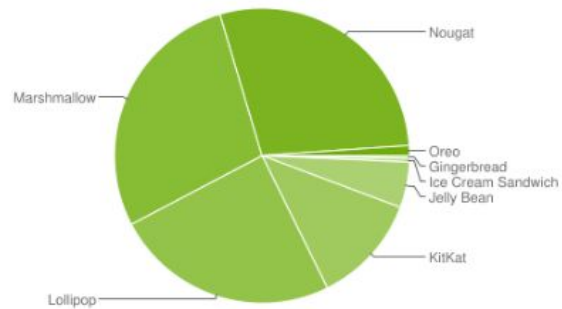
Operating System	2017 Units	2017 Market Share (%)	2016 Units	2016 Market Share (%)
Android	1,320,118.1	85.9	1,268,562.7	84.8
iOS	214,924.4	14.0	216,064.0	14.4
Other OS	1,493.0	0.1	11,332.2	0.8
Total	1,536,535.5	100.0	1,495,959.0	100.0

Source: Gartner (February 2018)

Figura 3.4. Ventas mundiales de teléfonos inteligentes a usuarios finales por sistema operativo en 2017 [15]

- El mayor porcentaje de los dispositivos móviles que poseen sistema operativo Android, cuenta con versión 5.0 o mayor, es decir, la versión a partir de la cual soporta Android Auto. Este porcentaje es aproximadamente el 80% según datos de sitios oficiales, en la *Figura 3.5* se describe estas estadísticas.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%



Datos recopilados durante un periodo de 7 días hasta 8/1/2018.
No se muestran versiones con una distribución inferior al 0,1%.

Figura 3.5. Estadísticas de versiones de la plataforma Android [16]

3.3.2 CarPlay

Es un estándar que Apple ha incorporado a sus dispositivos móviles iOS para que sean capaces de trabajar con los sistemas IVI, lanzado inicialmente en el año 2014.

El objetivo de este estándar es facilitar el acceso directo a la funcionalidad del dispositivo iOS, control y uso, directamente a través de los sistemas de control nativos del automóvil del fabricante.

Arquitectura

Para la comunicación entre el automóvil y el dispositivo móvil es necesario contar con protocolos de comunicación, un formato bien definido para que podamos transmitir información entre los dispositivos.

Una primera manera de conectar el sistema IVI con el dispositivo móvil es de manera cableada, por medio de cable USB. Tenemos en consecuencia un primer esquema, que podemos representar de la siguiente manera.

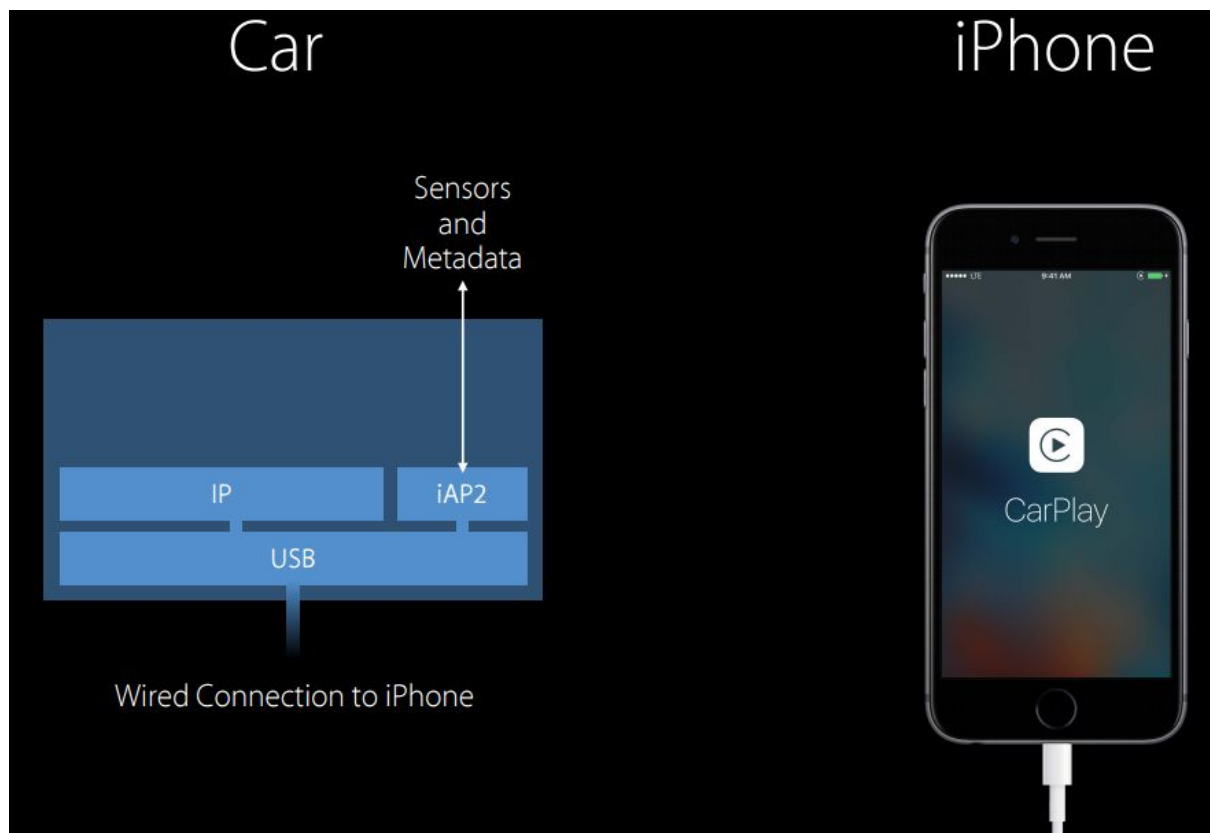


Figura 3.6. Conexión iPhone - Sistema IVI.

Esta arquitectura nos muestra la manera en que se conecta el iPhone con el sistema IVI del automóvil. Se cuenta con 3 componentes esenciales, el puerto USB, encargado de la conexión con el dispositivo móvil.

Por otro lado tenemos el protocolo iAP2 (también conocido como Apple Accessory Protocol), éste es usado para la comunicación entre dispositivos iOS y accesorios tales como estación de conexiones, adaptadores de automóviles, dispositivos HID (Human Interface Device).

Apple ha rediseñado el original protocolo iAP para agregar una nueva versión llamada iAP2 cuando fue introducido el conector de 9 pines liviano. Alwise, empresa dedicada a soluciones inalámbricas, implementa esta nueva versión del protocolo iAP para el Bluetooth. Las funcionalidades que encontramos entre otras son:

- Transferencia de información de metadatos.
- Transferencia de información de contactos.
- Enviar informe de HID.
- Recibir información de los medios.

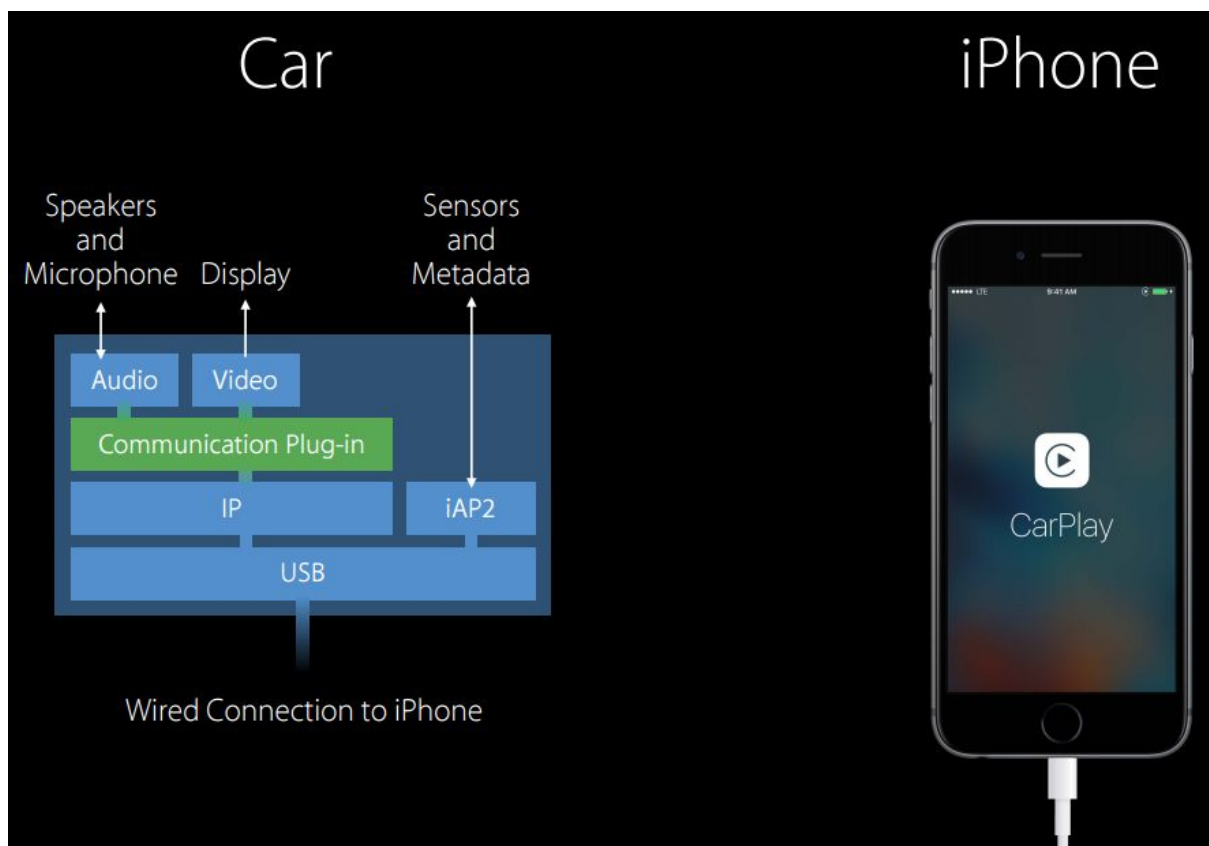


Figura 3.7. Arquitectura del Sistema IVI del automóvil incluyendo Audio y Video.

Tenemos la parte de Audio y Video, para Speakers and Microphone y Display respectivamente. Y el protocolo iAP2 para Sensors and Metadata. Estas son claves del requerimiento del vehículo. Pasamos a analizar uno por uno las anteriores claves.

- High Resolution Display:
 - Pueden ser de diferentes tamaños, que van de 800 x 480 a 1920 x 720. El tamaño de la pantalla física mínima es de 6 pulgadas.

- Los fabricantes de automóviles son los responsables de seleccionar el tamaño de la pantalla adecuada.
- Speaker and Microphone:
 - El audio principal es utilizado para multimedia, teléfono y Siri².
 - Entrada mono y salida estéreo.
 - El audio alternativo es usado para notificaciones, siempre es mezclado con el audio principal.
- Sensors and Metadata:
 - Velocidad de información.
 - GNSS (Global Navigation Satellite System).
 - Información de localización por satélite.
 - GPS y GLONASS³ requerido.
 - Sistema de navegación Galileo recomendado.
 - Conexión por cable USB
 - Wireless, WiFi + Bluetooth. En dos etapas.
 - En primera instancia se conecta el Bluetooth, luego, el automóvil envía las credenciales del WiFi al dispositivo móvil, a continuación se adquiere la conexión WiFi, y finalmente se establece la sesión de CarPlay bajo WiFi.

² Aplicación con funciones de asistente personal a veces con su propia personalidad para iOS, macOS, tvOS y watchOS.

³ Sistema Global de Navegación por Satélite (GNSS) desarrollado por la Unión Soviética, constituye el homólogo del GPS estadounidense y del Galileo europeo.

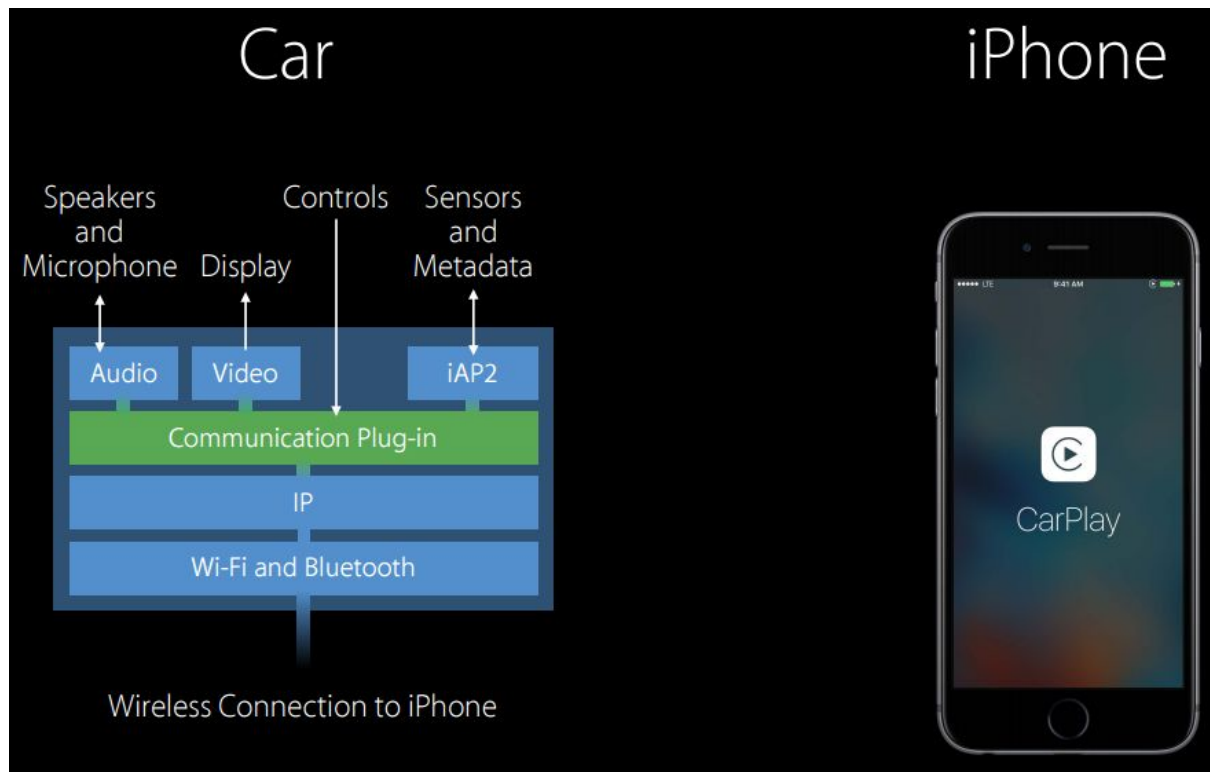


Figura 3.8. Arquitectura del Sistema IVI incluyendo Controles y Sensores.

Por último detallaremos la parte de la comunicación, que en la figura lo veremos reflejado en el ítem Controls. [17]

- Controls:
 - Touch:
 - La pantalla táctil es la interacción directa con la interfaz del usuario.
 - El automóvil envía las coordenadas **x** e **y**.
 - Táctil simple de alta o baja fidelidad. Alta fidelidad implica que el usuario pueda interactuar con la pantalla táctil mediante golpes y desplazamientos sobre la misma. Esto requiere sistemas de alta performance. Baja fidelidad por otro lado implica interacción con la pantalla a través de simples golpes individuales en la misma. Recomendado para pantallas táctiles resistentes.
 - Knobs and Controls:
 - Refieren a las perillas y controles provistas por el automóvil.
 - Las características necesarias son:

- Mando giratorio o touch pad.
- Atrás.
- Seleccionar.
- Voice:
 - Refiere a los botones incorporados en el volante del automóvil que nos permite a través del sistema de procesamiento de voz Siri, interactuar con el sistema IVI.

3.3.3 Mirrorlink

Es un estándar de interoperabilidad de dispositivos que ofrece la integración entre un dispositivo móvil y el sistema IVI de un automóvil. Permite que una aplicación alojada en el teléfono inteligente sea ejecutada en el mismo y mostrada en la pantalla del sistema IVI integrado en el automóvil, permitiendo al usuario interactuar con la aplicación a través de los controles que posee el volante, los botones del tablero o bien desde la pantalla táctil. [18]

Mirrorlink utiliza un conjunto establecido de tecnologías no propietarias, tales como IP, USB, WiFi, Bluetooth, protocolo de tiempo real (RTP, para el audio) y UPnP (Universal Plug and Play). Además, utiliza VNC (Virtual Network Computing) como el protocolo base para mostrar la interfaz de usuario de las aplicaciones del dispositivo móvil en la pantalla del sistema IVI, para su comunicación.

Es un estándar abierto de la industria para la integración Automóvil-Teléfono inteligente.

- Trata de proporcionar una guía común de desarrollo de aplicaciones para automóviles.
- Proporciona la infraestructura para que la fabricación de certificación de aplicaciones operen efectivamente.
- Trata de proporcionar transparencia a través del proceso de certificación para la conectividad y aplicaciones.

Arquitectura

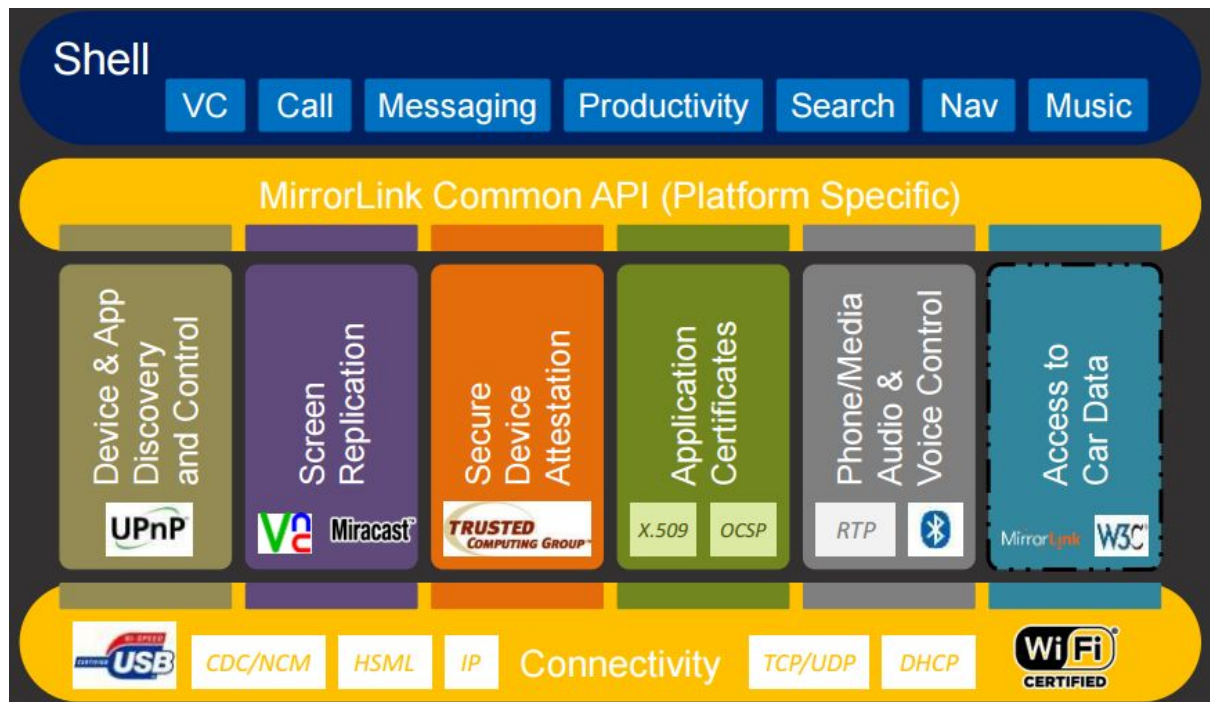


Figura 3.9. Arquitectura Mirrorlink.

La arquitectura está basada en estándares existentes. Pasaremos a analizar brevemente cada módulo de la misma. [19]

Connectivity: Realizar la conexión física. Puede ser a través de cable USB o con WiFi mediante tecnología P2P. La conectividad se logra involucrando diferentes métodos, entre los cuales encontramos, Ethernet, IPv4, ARP, TCP, UDP y DHCP.

Device & App:

- Device Discovery
 - Basado en UPnP (Universal Plug and Play) - SSDP (Simple Service Discovery Protocol).
 - Búsqueda desde el sistema IVI.
- Device Identification
 - El sistema IVI lee la descripción del teléfono inteligente.
 - Contiene los detalles del dispositivo y los fabricantes.
- Application Discovery

- El teléfono contiene un conjunto de aplicaciones Mirrorlink certificadas.
- Los sistemas IVI solicitan el conjunto de aplicaciones Mirrorlink disponibles (certificadas).
- El teléfono provee la lista de aplicaciones: Nombre, Descripción, URL, iconos, proveedor, categorías de las aplicaciones.
- El sistema IVI muestra al usuario las aplicaciones.
- Application Control
 - El usuario selecciona una aplicación desde el sistema IVI.
 - El sistema IVI solicita la ejecución de la aplicación seleccionada.
 - El teléfono ejecuta la aplicación seleccionada y lleva a esta a primer plano.
 - El teléfono retorna la URL a la interfaz de la aplicación de usuario.

Screen Replication:

- Visual
 - Remote UI:
 - Protocolo RFB (Remote Frame Buffer), es un simple protocolo para el acceso remoto a interfaces de usuarios gráficas.
 - Codificación FB.
 - Transparente a las aplicaciones.
 - Extensiones para temas visuales:
 - Tamaño y resolución de la pantalla.
 - Modo drive/park.
 - Modo día/noche.
- Eventos
 - Eventos simple-touch.
 - Eventos multi-touch.
 - Skinning support.

Phone/Media Audio & Voice Control

- Legacy Bluetooth

- BT HFP (Handsfree Profile), manos libres, es usado para permitir la comunicación con teléfonos móviles dentro del automóvil.
- BT A2DP (Advanced Audio Distribution Profile).
- RTP Audio Streaming

Application Certificates

Los usuarios instalan la aplicación desde cualquier App Store, y desde el dispositivo móvil se lo enlaza a la App Cert desde la base de datos de CCC⁴ (Car Connectivity Consortium). Luego el teléfono valida el certificado, y el mismo regularmente chequea el estado de la App Cert con la base de datos de CCC.

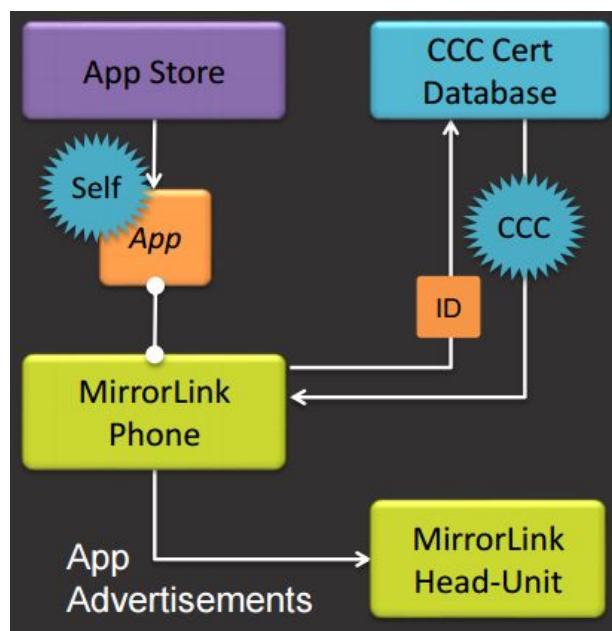


Figura 3.10. Ciclo de Certificación de una APP Mirrorlink.

Access to car data

- Common Data Bus
 - Nivel de servicios de datos.
 - Servicio de advertencias.
 - Servicio de inicio/parada.
- Service Binary Protocol
 - Nivel de servicios de objetos.

⁴ Es una organización desarrollada por Mirrorlink® que impulsa las tecnologías globales, para soluciones de conectividad de automóviles centradas en teléfonos inteligentes.

- Valor de objetos.
- Tipos de objetos (Integer, Double, struct, []).
- Acceso a los objetos (Get, Set, Subscribe).

3.4. Selección de la tecnología para el desarrollo de Steer

Las tecnología seleccionada para el desarrollo de la aplicación fue MirrorLink, analizaremos cuales fueron los motivos de la decisión.

Como primera acción, al ser un software especializado para sistemas IVI de los automóviles, la decisión iba a tomarse en base a la disponibilidad de herramientas que teníamos para el desarrollo. La misma surgían a partir de tres diferentes estándares, CarPlay, Android Auto y Mirrorlink, que hemos explicado detalladamente en este capítulo, cada uno con su arquitectura, ventajas y desventajas.

Por aspectos prácticos en principio se descartó CarPlay, de Apple, ya que implicaba abordar un lenguaje de programación para hacer el software, tal como Objective-C o bien Swift, que, al no poseer conocimientos sólidos de los mismos, nos implicaba un esfuerzo extra aprender los frameworks necesarios y el lenguaje en sí para el desarrollo de la aplicación y tampoco se contaba con un dispositivo móvil para realizar pruebas. Además presentaba características muy similares a Android Auto. Las alternativas más factible fueron por ende, la de utilizar Android Auto o Mirrorlink, ambas en Android, con la ventaja de poder utilizar Java como plataforma de desarrollo del cual si poseíamos conocimientos previos y nos facilitaba el proceso de elaborar el código.

Android Auto fue en principio una opción atractiva ya que contabamos con todas las herramientas que se necesitaban disponibles para su uso. No obstante esta API solo permite realizar un formato de aplicaciones muy acotado comparado con lo que se pretendía realizar la solución a la problemática a la cual planteamos.

La principal diferencia entre los tres protocolos que nos llevó a decidirnos por MirrorLink, más allá de los aspectos técnicos, fueron las posibilidades y limitaciones de las aplicaciones que permite desarrollar. En MirrorLink, la aplicación tiene control de toda la pantalla de la unidad central, por lo que la aplicación tiene la posibilidad de enviar cualquier información a la pantalla. Por el contrario, Android Auto solo permite aplicaciones que reproducen medios de audio o implementan servicios de mensajería y no toman el control de toda la pantalla, sino que solo proporcionan el contenido. En este contexto, CarPlay es similar a Android Auto, pero no tan claramente restringido a esas dos clases de aplicaciones. Por lo tanto, no es posible, por ejemplo, desarrollar una aplicación que proporcione un sistema de navegación alternativo, ya que Google y Apple sólo permitirían

que sus soluciones de navegación nativas (es decir, Google Maps o Apple Maps respectivamente) formen parte de la solución, mientras MirrorLink es más abierto.

También en los tres protocolos, se deben seguir estrictamente las pautas exigidas por los creadores del protocolo para garantizar que todos los problemas relacionados con la seguridad se resuelvan y no representen un riesgo para la seguridad del conductor. Los tres protocolos poseen procesos de certificación / aprobación de aplicaciones. En MirrorLink, una aplicación puede estar certificada solo para "modo aparcado" o también para "modo de conducción". Obviamente, la certificación de "modo de conducción" tiene mayores requisitos en cuanto a la seguridad del conductor que una simple certificación de "modo aparcado".

Finalmente se concluyó que Mirrorlink era la alternativa ideal para realizar el prototipo de la aplicación. A diferencia de Android Auto, con Mirrorlink se puede realizar una aplicación Android sin límites en cuanto a uso de herramientas provistas para la plataforma, es decir, que se podía desarrollar cualquier tipo de aplicación sin mayores problemas. Otro punto a favor, fue que nos proveía un emulador para simular un sistema IVI compatible con Mirrorlink, que sería la réplica exacta de un automóvil que contiene el dispositivo con dicha tecnología. Esto sería de gran ayuda ya que no se cuenta con vehículo con estas características.

Capítulo 4

Propuesta de desarrollo

4.1 Justificación de la aplicación

En los capítulos anteriores hemos observado cómo la utilización de dispositivos móviles en los últimos años ha crecido de manera exponencial; la sociedad ha adoptado su uso como un hábito diario y, año tras año esto se reafirma cada vez más, todo esto nos da la pauta que, trabajar en esta área de la tecnología puede darnos resultados fructíferos y eventualmente abrir puertas para proyectos futuros en el sector que investiguemos y trabajemos. En efecto, durante los últimos años el uso del dispositivo móvil se ha llevado en todo tipo de terrenos, uno de los implicados es el automóvil. Esto se ha vuelto común y cotidiano, a modo tal, que es un gran problema para los principios que se plantea para un conductor de automóvil, que radica básicamente, en la no distracción del mismo mientras se está manejando el vehículo.

No obstante, si bien el concepto de ciudades inteligentes evoca imágenes de rutas repletas de sensores, mapas en vivo de la red de subtes y centros de control llenos de enormes pantallas de datos, también podemos tomar ventaja de los sensores que millones de personas ya llevan consigo, incorporados en sus teléfonos inteligentes, y reducir considerablemente el costo de lo que valdría construir un centro de gestión de tráfico, y los servicios, conocimiento y análisis de datos necesarios.

Si tomamos como ejemplo la ciudad de Ghent en Bélgica, observamos cómo podemos acercarnos al futuro de la infraestructura "inteligente" sólo mediante el uso de un par de aplicaciones para smartphones. La ciudad belga es uno de los primeros en Europa occidental en inscribirse al programa Waze Connected Citizens, donde las autoridades de transporte de la ciudad intercambian datos de tráfico e incidentes con la aplicación de navegación, propiedad de Google. Usuarios de Waze pueden reportar los accidentes y los atascos de tráfico, que luego se transmiten a los demás conductores. Los datos también son compartidos con los gobiernos locales para ayudar con la gestión del tráfico en tiempo real, incluyendo el envío de respuesta a emergencias. A su vez, otras 55 ciudades que adoptaron este esquema de Waze dan información sobre los cierres de carreteras y nuevas políticas de tráfico.

En otras ciudades, las autoridades municipales y aplicaciones móviles ya colaboran.

Citymapper, una aplicación para el transporte urbano fue desarrollada inicialmente en Londres utilizando los datos abiertos de la ciudad proporciona alrededor de líneas de metro, autobús y tren. Además de proporcionar asesoramiento de transporte al público en general, Citymapper ofrece herramientas de gestión de datos de las autoridades de tránsito que ayudan a las alertas de retroalimentación y incidentes de los usuarios.

Del mismo modo, Strava, una aplicación de seguimiento de ciclismo, vende datos anónimos a ciudades como Londres u Orlando, Florida. La gente está ayudando a planear las rutas de los carriles de bicicleta y a su vez es una medida para mejorar la congestión de tráfico. Uber, la red de transporte a particulares más grande del mundo, también ha comenzado a compartir los datos del conductor con funcionarios en ciudades como Boston, para ayudar a crear planes o políticas de reducción de la congestión. Datos similares, incluyendo los sitios de subida y bajada de pasajeros (anónimos) y la hora, también son proporcionados por los servicios de taxi.

El análisis de estos datos de la ciudad de Nueva York en 2014, por investigadores de la Universidad de Cornell, encontraron que la flota de taxis de la ciudad podría reducirse hasta en un 40 por ciento si más habitantes optarán por compartir taxis. A partir de ese estudio, el Senseable City Lab ha estado trabajando con Uber en la optimización de su servicio de viajes compartidos Uber Pool.

Actualmente esto se puede lograr gracias a los grandes avances tecnológicos en el área automovilística que hubo en los últimos años, informatizando gran cantidad de partes del mismo y, por otro lado, el inmenso avance tecnológico de los dispositivos móviles en cuanto a funcionalidades, y compatibilidades con otras herramientas; se propone desarrollar una aplicación que funcione combinando estas dos temáticas fundamentales, con el propósito de facilitar o enriquecer a la conducción del automóvil, con la premisa de no distraer al conductor que utiliza la misma y siguiendo las normativas adecuadas para realizar el desarrollo del mismo.

En colaboración con el CeSPI, se busca adaptar a estas nuevas tecnologías, los servicios que han sido desarrollados para nuestra ciudad, como SEM o CENIT, de manera tal de dar un servicio extra, utilizando las herramientas venideras en el área de ciudades y dispositivos inteligentes.

Steer entonces, es una aplicación móvil de navegación, diseñada para ser comandada directamente desde el tablero del automóvil (Sistema IVI) donde, el conductor dispondrá de un mapa de seguimiento de su vehículo (navegación), y podrá ver o denunciar distintos eventos y alertas de la vía pública por medio del sistema CENIT, mencionado

anteriormente y, la posibilidad de iniciar/finalizar el estacionamiento, por medio del sistema SEM, también nombrado de manera reciente.

En el próximo capítulo se analizará en detalle todo lo que respecta a arquitectura y funcionalidad de la aplicación, que es el tema principal del presente proyecto, con ello se podrá comprender desde las decisiones tomadas, los problemas encontrados, las soluciones a ellos y las especificaciones del trabajo concluido.

4.2 MirrorLink

MirrorLink ha sido el estándar elegido como protocolo de conexión entre el teléfono y el sistema IVI del automóvil, veremos cómo se compone una aplicación para integrar esta tecnología y las herramientas que fueron usadas.

Podemos ver a este estándar, como una herramienta que nos permite unir tres sectores diferentes del área tecnológica para cumplir un objetivo en común.

Teléfono inteligentes: El Servidor MirrorLink será donde se va a ejecutar la aplicación. Para ello, el sitio oficial de MirrorLink provee un listado completo de los modelos de celulares que han adoptado este protocolo entre sus interfaces de conexión; este sitio se va actualizando de manera permanente. Sin este dispositivo las aplicaciones no van a ser reconocidas por el celular como Mirrorlink App, por ende no va a ser reconocida por el cliente, y en consecuencia no va a funcionar. Es por ello que es importante contar con esta opción en el dispositivo. Cabe aclarar que, al principio había una disponibilidad baja de este estándar en los teléfonos celulares, pero, a medida que pasó el tiempo, esta compatibilidad ha ido creciendo de manera vertiginosa, es por ello que hoy ya un gran número de marcas adoptan esta tecnología como algo nativo y muchas nuevas están siendo incluidas a medida que pasa el tiempo.

Es por ello que para nuestro desarrollo de la aplicación móvil, fue fundamental contar con un dispositivo móvil que tenga esta tecnología para poder realizar las pruebas necesarias de la aplicación.

Sistemas IVI: Los Clientes MirrorLink, al igual que los teléfonos son dispositivos comerciales que, mediante convenios con la CCC, han implementado el protocolo para la conexión. Estos, son ofrecidos por las empresas automotrices que incorporan a los dispositivos del vehículo, o por empresas fabricantes de este tipo de auto partes.

Para poder realizar pruebas, MirrorLink facilita un simulador para utilizar en una computadora, del cual ahondaremos más adelante.

Aplicaciones: Finalmente, tenemos el elemento principal del proyecto, que es la

aplicación Android propiamente dicha. Haremos hincapié sobre este punto, dado que ha sido el eje principal del trabajo.

Los pasos básicos para integrar MirrorLink en una aplicación Android son,

- Declarar el permiso para el uso de MirrorLink.
- Definir las acciones para iniciar y terminar la aplicación desde el dispositivo IVI que es la encargada de iniciar la aplicación cuando se conecta el teléfono.
- Modifique la IU para que sea adecuada para usar mientras conduce.
- Registre la aplicación con CCC e inicie el proceso de certificación MirrorLink.

MirrorLink Android Studio Plugin

Este plugin nos ayuda a comenzar o adaptar un proyecto a través de una interfaz de usuario creada por la misma CCC.

El plugin funciona como una plantilla que nos permite crear un nuevo proyecto, en Android Studio, de tipo MirrorLink. Al hacerlo se declaran automáticamente los permisos para el uso de MirrorLink, y las acciones para iniciar y terminar la aplicación desde el dispositivo IVI, que es la encargada de iniciar la aplicación cuando detecta que se se conecta un teléfono.

Por otro lado para poder ejecutar la aplicación, en principio a modo test, se necesita generar un archivo de certificación único para cada teléfono y para cada versión de la aplicación. El plugin automatiza la creación de este certificado, regenerandolo en cada compilación.

MirrorLink Client Simulator

Para facilitar las pruebas de la aplicación la CCC proporciona un simulador de Cliente MirrorLink.

Esta herramienta simula un dispositivo IVI, con protocolo MirrorLink, de un automóvil en la computadora, lo cual es bastante beneficioso para probar una aplicación sin necesidad de conectarlo directamente al automóvil.

Este paso tiene requisitos para asegurar su funcionamiento, tales como un sistema operativo en particular (Ubuntu), cierta versión del mismo, cierta versión de kernel, ciertas versiones de librerías de Python necesarias para generar la interfaz gráfica del simulador. Esta información puede variar ya que la documentación nos provee los ambientes que fueron probados y verificados como correctos para la ejecución del simulador, pero también puede haber variación en la versión de algunos componentes. La instalación del software es

uno de los pasos más importante de para recrear las pruebas de la App, ya que va a ser el que nos va proveer el ambiente que simula una unidad IVI real.

Otro punto que definimos para la creación de aplicaciones MirrorLink es la adaptación de la interfaz de usuario. La CCC ofrece una serie de requerimientos y recomendaciones en este punto.

Cualquier tipo de aplicación se puede adaptar para su funcionamiento con MirrorLink, pero no todas las aplicaciones se encuentran aptas para su uso mientras se conduce. Para hacer una aplicación que se puede utilizar mientras se conduce, la interfaz de usuario no puede crear distracciones para el conductor. Vale aclarar que las restricciones son solo para las aplicaciones se van a utilizar en modo conducción. El concepto fundamental es limitar la atención e interacciones necesarias del conductor. El sitio oficial del CCC se encuentra mucha información, entre la que podemos localizar la específica para crear este tipo de aplicaciones y las diferentes normativas que nos sugieren la idea de una aplicación apta para el uso en dispositivos IVI.

Una característica importante de la API MirrorLink es que, cuando el teléfono está conectado al sistema IVI, notifica el cambio de estado si el auto está estacionado o en movimiento. Con esto se puede decidir no poner toda la funcionalidad a disposición en el modo conductor pero si cuando se está estacionado o no se está conectado.

Los conceptos generales que son la base de los requisitos de diseño son los siguientes:

- **Mostrar texto mínimo:** No mostrar una gran cantidad de mensajes de texto o que sean complejos
- **No usar la entrada de texto:** Aunque esté permitido en algunas regiones, es una mala práctica realizar esta operación durante el modo de manejo. Se debería usar el modo de estacionamiento para la entrada de texto o la entrada de voz si es posible.
- **Use el modo de estacionamiento para la funcionalidad adicional:** Como una aplicación MirrorLink sabe si está conectado a la unidad IVI o en el modo de estacionamiento, puede ofrecer características adicionales o información al controlador mientras está parado. Cuando el automóvil entra en el modo de conducción, cambia a la interfaz de usuario más simple.
- **Usar alto contraste:** La IU debe ser legible en una variedad de condiciones de iluminación, incluso bajo la luz solar directa. Para lograr esto, todas las combinaciones de texto deben tener un nivel de contraste mínimo.

- Usar Letras grandes: se requieren tamaños de letra grandes para que se puedan leer fácilmente en todas las pantallas. Use una fuente estándar de Android para una buena legibilidad. Tenga en cuenta que el tamaño mínimo se medirá en la pantalla del dispositivo IVI.
- Use botones grandes: los botones deben ser lo suficientemente grandes como para poder usarlos durante la unidad. La longitud mínima del borde para un botón es de 10 mm con un área de 200 mm cuadrados.

El último proceso para que la aplicación sea compatible con MirrorLink fuera de un ambiente de prueba es la certificación.

Dado que el uso de una aplicación mientras se conduce afecta la seguridad del conductor, los pasajeros y otras personas, CCC tiene un proceso para verificar que la aplicación funcione correctamente y cumpla con las pautas de UI. El proceso de certificación comienza con el registro de su aplicación en CCC. Se debe seleccionar, de un lista, uno de de los laboratorios autorizados y gestionar las pruebas. CCC no cobra por el proceso de certificación, pero hay tarifas asociadas con las pruebas en el laboratorio.

Hay un formulario clave que deberá completar para el proceso de certificación. Esto se conoce como la declaración de conformidad de implementación del protocolo de aplicación (aplicación PICS) . La aplicación PICS es una hoja de verificación en la que enumera todas las funciones compatibles de su aplicación y garantiza que el laboratorio pueda probarla correctamente.

Se trabaja en conjunto con el laboratorio para completar las pruebas y proporcionar un informe al CCC. Una vez que se proporcione un informe de prueba exitoso, será certificado después de una revisión administrativa. Finalmente se puede publicitar su aplicación como compatible con MirrorLink.

Si se realizan cambios en su aplicación que afecten la funcionalidad de MirrorLink o el diseño de la interfaz de usuario en modo conducción, se deberá notificar a CCC para actualizar la certificación a la última versión de la aplicación.

4.3 Herramientas utilizadas

Google Play Services

Esta librería se utiliza para obtener la localización geográfica. Si bien el SDK general de Android cuenta con funciones de localización, actualmente se recomienda utilizar las de Google Play Services que proporciona algunas facilidades y mejoras, entre ellas, la más importante es que no necesitamos indicar de forma explícita el proveedor de localización a utilizar (señal móvil, Wi-Fi, o GPS) sino que ofrece un proveedor propio (Fused Location Provider) que se encarga automáticamente de gestionar las fuentes disponibles para obtener el dato de la localización.

Se debe destacar que no se cuenta con un método que devuelve la posición actual de manera explícita, ya que el tiempo que se tarda en obtenerla es impredecible. La estrategia que se utiliza consiste en indicar al sistema, en primer lugar y según nuestros requerimientos, la precisión y periodicidad con la que se necesita recibir las actualizaciones de la localización actual, finalmente definimos un método encargado de procesar los datos recibidos.

Google Maps APIs

Las Google Maps APIs son un conjunto de APIs ofrecidas de forma gratuita con las que podemos agregar mapas basados en datos de Google Maps a la aplicación, además de muchas otras herramientas para agregar funcionalidad a la aplicación y enriquecer el aspecto visual. [20]

Entre todas las características ofrecidas por estas APIs, podemos resumir las más importantes para nuestro proyecto:

Google Maps Android API: Administración en forma automática del acceso a servidores, descargas de datos, visualización de mapas y respuesta a gestos de mapas de Google Maps.

Agregar marcadores, polígonos y superposiciones a un mapa básico, y para cambiar la vista del usuario de modo que se muestre un área del mapa en particular. Estos objetos proporcionan información adicional de ubicaciones en el mapa y permiten la interacción del usuario con este.

Permite cambiar el estilo del mapa, ya que, si bien el aspecto de Google Maps es conocido y confiable para los usuarios de todo el mundo, si se necesita algo un poco diferente, las APIs de Google Maps le permiten adaptar el estilo del mapa a sus demandas específicas. Puede simplificar el mapa para centrar la atención en sus datos, o bien modificar el estilo para que combine mejor con el resto de su aplicación.

Google Maps Geocoding API: Es un servicio web que proporciona codificación geográfica normal e inversa que permiten acceder a la base de datos global de direcciones de Google, de modo que podrá buscar ubicaciones en el mapa a partir de una dirección, o bien buscar la dirección de un lugar determinado en cualquier parte del mundo. Además con autocompletado de direcciones se facilita el ingreso de las direcciones a los usuarios. Cualquier campo de texto se puede mejorar con la potencia del autocompletado automático de Google Maps, que ofrece un modo rápido, preciso y sencillo de ingresar direcciones.

Google Maps Directions API: Servicio web con el cual se puede armar una ruta con instrucciones para llegar a un destino deseado utilizando el potente motor de rutas de Google. La ruta contempla datos actuales de tráfico para ofrecer la mejor, pudiendo aplicar un código de colores a las calles principales para reflejar el volumen actual de tráfico en tiempo real. También se ofrecen rutas alternativas, y los usuarios pueden arrastrar las rutas sobre el mapa para hacer cambios. Las rutas pueden evitar los peajes o las autopistas, y es posible reducir la duración del recorrido calculando el orden óptimo en el que conviene visitar cada ubicación. Sumado a esto se puede predecir la duración del viaje, estimando en función de los datos históricos por horario y día de la semana.

Google Maps Places API: Con esta API podemos acceder a la base de datos de Google Maps la cual cuenta con millones de millones de direcciones y puntos de interés.

Es utilizada para ayudar al usuario en la búsqueda de direcciones, que luego serán utilizadas para el cálculo de la ruta de navegación, ya que cuenta con la función de autocompletar mientras se escribe la dirección.

Entre sus funciones principales tenemos:

- Búsquedas de sitios: Devuelve una lista de sitios sobre la base de la ubicación de un usuario o un texto de búsqueda.
- Las solicitudes de detalles del sitio: Devuelven información más detallada sobre un sitio específico, incluidas las reseñas del usuario.
- Agregado de sitios: Permite complementar los datos de la base de datos de Google Places con datos de tu aplicación.
- Fotos de sitios: Proporciona acceso a las millones de fotos relacionadas con los sitios almacenadas en la base de datos de Google Places.

- Autocompletado de sitios: Se puede usar para completar automáticamente el nombre o la dirección de un sitio mientras se escribe.
- Autocompletado de consultas: Se puede usar para proporcionar un servicio de predicción de consultas para búsquedas geográficas basadas en texto, al devolver consultas sugeridas mientras se escribe.

Google Maps Roads API: Contiene el servicio “snap to road” que devuelve la mejor aproximación de geometría de rutas para un conjunto específico de coordenadas de GPS. Este servicio admite hasta 100 puntos de GPS recopilados a lo largo de la ruta y devuelve un conjunto de datos similar con los puntos ajustados a las rutas más probables (a través de Snap to Roads) que recorrió el vehículo. También puedes solicitar que se interpolan los puntos, lo que resultará en un trayecto que siga sin dificultad la geometría de la ruta. Con esto podemos evitar que, al dibujar la posición del usuario en el mapa, se sitúe fuera del camino ya que la posición puede tener un margen de error.

Firestore

Es una plataforma de desarrollo para aplicaciones web o móviles subida en la nube y disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptarla a medida de sus necesidades.. Proporciona gran parte de la infraestructura de servidor necesaria como:

- Base de datos en tiempo real (Realtime database)
- Sistema de autenticación (Authentication)
- Mensajería y Notificaciones (Cloud Messaging)
- Almacenamiento (Storage)
- Estadísticas de uso (Analytics)
- Reporte de errores (Crash Reporting)
- Publicidad (AdMob)

Para nuestras aplicaciones utilizaremos los servicios de base de datos y sistema de autenticación.

Firestore proporciona un servicio de base de datos NoSQL alojada en la nube con la particularidad de ser en tiempo real, esto implica que cualquier cambio realizado en los datos por cualquier cliente (usuario, aplicación, dispositivo) se sincronizará

automáticamente y de forma inmediata (siempre que la conexión lo permita) en el resto de clientes, sin necesidad de que éstos vuelvan a consultar los datos. No es el modelo de base de datos tradicional, con tablas y registros estructurados, sino un modelo de datos jerarquizados similar a la estructura de un archivo JSON, es decir, en forma de árbol donde cada nodo puede contener un valor o bien contener nodos hijo, que a su vez podrán contener valores o tener nuevos nodos hijo (hasta un máximo de 32 niveles de anidación).

Una de las principales ventajas para su uso en aplicaciones móviles es que está preparado para permitir interactuar con la base de datos cuando el dispositivo no tiene conexión (siempre dentro de unos límites) mediante un sistema de cachés y colas de escritura locales. Cuando el dispositivo vuelve a tener conexión, los cambios locales serán sincronizados automáticamente con la base de datos y, si aplica, con el resto de clientes conectados a ella [21].

La plataforma Firebase provee un servicio de autenticación para facilitar la creación de sistemas de autenticación seguros y compatible. Facilita la creación de un sistemas autenticación seguros proporcionando una solución de identidad de extremo a extremo, compatible con cuentas de correo electrónico/contraseña, autenticación telefónica, Google, Twitter, Facebook, entre otros. Fue creado por el mismo equipo que desarrolló Google Sign-in, Smart Lock y Chrome Password Manager, y aplica la experiencia interna de Google en la administración de una de las bases de datos de cuentas más grandes del mundo. También incorpora una interfaz de usuario abierta y personalizable que implementa recomendaciones para la autenticación en sitios web y dispositivos móviles. Se puede configurar todo el sistema de autenticación de una aplicación con pocas líneas de código, incluso con la capacidad de administrar casos complejos como la fusión de cuentas. [22]

Bootstrap

Es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Fue utilizado para facilitar el frontend de la aplicación web, ya que nos otorga todas estas ventajas para un diseño rápido y sencillo. Bootstrap es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores.

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS⁵ que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada bootstrap.less incluye los componentes de las hojas de estilo. El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados. [23]

Eclipse

Es un entorno de desarrollo multiplataforma, conformado a modo de “una base” que necesita de plugins en función del lenguaje de programación que vayamos a utilizar, puede ser utilizado para el desarrollo de aplicaciones en Java, Python, C, C++, entre otros. Eclipse dispone de un Editor de texto con un analizador sintáctico y la compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests y refactorización. Para el desarrollo del backend de la aplicación web es ideal el uso de esta herramienta, por su simplicidad y compatibilidad que ofrece con el lenguaje Java, que fue el elegido como base para la elaboración del sitio. [24]

AngularJS

Es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC) con el objetivo de lograr mejor interactividad y experiencia de usuario. La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas⁶ de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos. [25]

Google App Engine

⁵ **Less** es un dinámico lenguaje de hojas de estilo que puede ser compilado en Hojas de estilo en cascada (CSS) y ejecutarse en el lado del cliente o en el lado del servidor

⁶ Las directivas son marcas en los elementos del árbol DOM, en los nodos del HTML, que indican al compilador de Angular que debe asignar cierto comportamiento a dichos elementos o transformarlos según corresponda.

Es un servicio de alojamiento web que presta Google, con la posibilidad de utilizarlo de forma gratuita. Este servicio permite ejecutar aplicaciones sobre la infraestructura de Google. Si no se cuenta con un dominio propio, Google proporciona uno con la siguiente estructura, *midominio.appspot.com*. También permite implementar un dominio propio a través de Google Apps. Actualmente se divide en dos módulos:

- Google App Standard Environment: Tiene posibilidad de programación acotada, entre los lenguajes que son compatibles con este módulo, encontramos Java, Python, PHP y GO. No Permite accesos a disco usando esto lenguajes, a cambio, los costes son más económicos.
- Google App Flexible Environment: Puede implementar cualquier lenguaje de programación mediante contenedores Docker⁷, permitiendo mayor control sobre la aplicación. Los costes son proporcionales al consumo de CPU.

Nuestro caso fue, utilizar esta herramienta para tener en una nube subida la aplicación web y simular su utilización en un navegador web real, para así, tener una aproximación de su uso que no sea el de correrlo de manera local. Se utilizó el lenguaje Java y Eclipse como software para integrar Google App Engine y realizar los deploys a la nube.[26]

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android que se basa en IntelliJ IDEA, otro IDE para JAVA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de las aplicaciones para Android, como las siguientes: [27]

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y frameworks de prueba.

⁷ Proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software.

- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

La primera versión estable fue publicada en diciembre de 2014, por lo que nos pareció una herramienta nueva e interesante para tratar el desarrollo de la aplicación mobile. Está disponible para múltiples plataformas, lo cual le da versatilidad para poder realizar la aplicación desde diferentes entornos, entre ellos encontramos Windows 2003, Vista, 7, 8, y 10, tanto plataformas de 32 como de 64 bits, GNU/Linux, Linux con GNOME o KDE y 2 GB de memoria RAM mínimo y macOS, desde 10.8.5 en adelante.

Sistema de Estacionamiento Medido (SEM)

El SEM, permite la autogestión de estacionamiento medido y pago por parte del usuario automovilista. Provee una API para poder integrar una aplicación móvil a los servicios que dispone el sistema, de manera tal de poder gestionar el estacionamiento a través de esta nueva aplicación.

Los principales servicios que utilizamos en el desarrollo de Steer son: iniciar estacionamiento, finalizar estacionamiento y consultar estado actual del usuario.

El servicio *iniciar estacionamiento* nos permite enviar una serie de parámetros que son procesados en el sistema de SEM y nos devuelve un conjunto de datos con el resultado de la operación ya sea que fue exitosa o no. De manera análoga al servicio recién descrito funciona el de finalizar el estacionamiento, se envían un conjunto de parámetros y se obtienen datos para luego procesar desde la aplicación móvil. Por último tenemos el servicio de consultar el estado actual del usuario conductor, que este nos proveerá información para cuando se necesite chequear todo lo relativo al usuario final, el pedido de la información al sistema de SEM funciona de manera equivalente a los servicios anteriores.

CENIT

Esta solución funciona como un centro de información general cuyo objetivo es gestionar eventos de distintos tipos (reclamos, incidentes, alarmas y notificaciones) que pueden ser generados por diversos canales de comunicación y moderados en forma manual o automática a través de un motor de inferencias. Provee una API para integrar sus

servicios a una aplicación móvil, cuya finalidad es la de tener la posibilidad de reportar eventos desde esta aplicación, para su posterior validación y visualización.

Steer integró de la gran cantidad de funciones disponibles, sólo un puñado de ellos para interactuar con el sistema de CENIT. Alertar un evento es uno de los servicios utilizados desde aplicación móvil, el cual consiste en enviar una serie de parámetros entre el que se encuentra el tipo de evento que será, bacheo, manifestación, luego los datos son enviados para que el sistema del backend procese dicha información y devuelva una respuesta del resultado del mismo. Otro servicio utilizado es el de listar los eventos; esta función nos brinda datos referentes a eventos alrededor de dónde se encuentra ubicado de manera geo-referencial el usuario.

Capítulo 5

Steer, el prototipo

5.1 Introducción

En este capítulo vamos a explicar las decisiones tomadas en cuanto a la arquitectura, diseño y funcionalidades de Steer, la aplicación desarrollada para esta tesina. La misma cuenta con dos versiones, una web y una móvil, y como su nombre lo indica, la primera de ellas se accede a través de un navegador web y la segunda a través de un teléfono inteligente que puede operar sobre el panel del control del automóvil. Ambas aplicaciones están nutridas desde la misma base de datos, ya que consumen los mismos servicios Rest, por lo que los datos administrados por ambas serán siempre consistentes. La versión móvil es la que se va usar mientras se está conduciendo, brinda funcionalidades como estacionamiento medido automático, seguimiento de autos, y también es un medio para alimentar el sistema. La versión web otorga mayor flexibilidad y robustez, está destinada especialmente para funciones administrativas y de control, esto incluye tareas en la que es necesario estar focalizado en la pantalla, escribir información de manera manual al sistema, ya sea para consultar o actualizar información.

5.2 Diseño de la aplicación

5.2.1 Arquitectura del proyecto

El proyecto en sí es un conjunto de múltiples sistemas comunicados entre sí, dónde, básicamente contamos con una aplicación móvil y una web las cuales se conectan a una misma base de datos y diferentes servicios REST.

La aplicación móvil Steer, es la que va a establecer la conexión con el sistema IVI del automóvil por lo tanto cumple un rol fundamental en la estructura del proyecto. Fue diseñada con una lógica minimalista, con pocos requerimientos de interacción y con una interfaz de voz de entrada para evitar en lo posible el tipeo.

Luego tenemos la aplicación web de Steer que va a brindar al usuario información acerca de los contactos del mismo y la manipulación de este tipo de información, además

de la posibilidad de ver los diferentes tipos de eventos en el mapa. Cuenta con una interfaz similar a la aplicación móvil, el frontend, que va a ser accesible desde un navegador web, con el cual el usuario final va a poder interactuar para consultar la información del sistema, provista por el backend, que realizará todo el procesamiento pesado de la aplicación web, la comunicación con los servicios y las consultas si es necesaria de las bases de datos.

Tanto la aplicación móvil como la web de Steer, se comunican con el sistema de alerta de eventos a través de los servicios prestados, de manera tal que los datos se mantengan íntegros. Todo este conjunto de sistemas interactúan entre sí para que la aplicación móvil se comporte de manera fluida y la experiencia del usuario final sea óptima.

Al ser la aplicación móvil el eje principal de este proyecto, necesitamos captar y procesar datos desde el dispositivo móvil, entre los cuales requerimos cubrir al menos las siguientes funciones:

- Captar a través de geolocalización la posición en la que se encuentra el usuario.
- Renderizado de diferente tipo de información con la que se interactúa.
- Notificación de acciones en tiempo real.

La aplicación móvil se implementó para Android y hace uso de las siguientes tecnologías/librerías:

- Se utilizó Android como Sistema Operativo de desarrollo como eje fundamental de la aplicación.
- Mirrorlink, como estándar de interoperabilidad entre el dispositivo móvil y el automóvil.
- Geofences, para facilitar la geolocalización dentro de áreas predefinidas.
- Firebase, como API para sincronizar datos en la nube en tiempo real, desde la aplicación.

Por otra parte, para la aplicación web de Steer se utilizó:

- HTML5, AngularJS y Bootstrap como tecnologías para el frontend y Java EE como tecnología de backend.

Ambas versiones consumen servicios de una aplicación existente para visualizar los eventos registrados y validados en la vía pública. Para ello, Steer:

- Implementa la comunicación con el servidor a través de servicios RESTful.
- Consume servicios provistos por el CeSPI (Centro Superior para el Procesamiento de la Información) para la comunicación con los servicios de estacionamiento medido, y CENIT.

Todas las herramientas con las que se ha trabajado en el transcurso de este proyecto son Open Source, esto implica múltiples características que nos parecen realmente interesantes. En principio que sea libre nos aporta transparencia, con esto queremos decir que tenemos acceso al código fuente para verlo y modificarlo añadiendo diferentes funcionalidades o bien cambiando el comportamientos del mismo. A su vez también nos ofrece gratuidad, esto se refiere a que tenemos la posibilidad de acceder al código y usarlo de manera libre y gratuita. Respecto a la seguridad, al disponer el código fuente en su totalidad, éste puede ser analizado por terceros en busca de fallos de diseño o de implementación, lo cual aumenta las posibilidades de encontrar algún posible bug en el código.

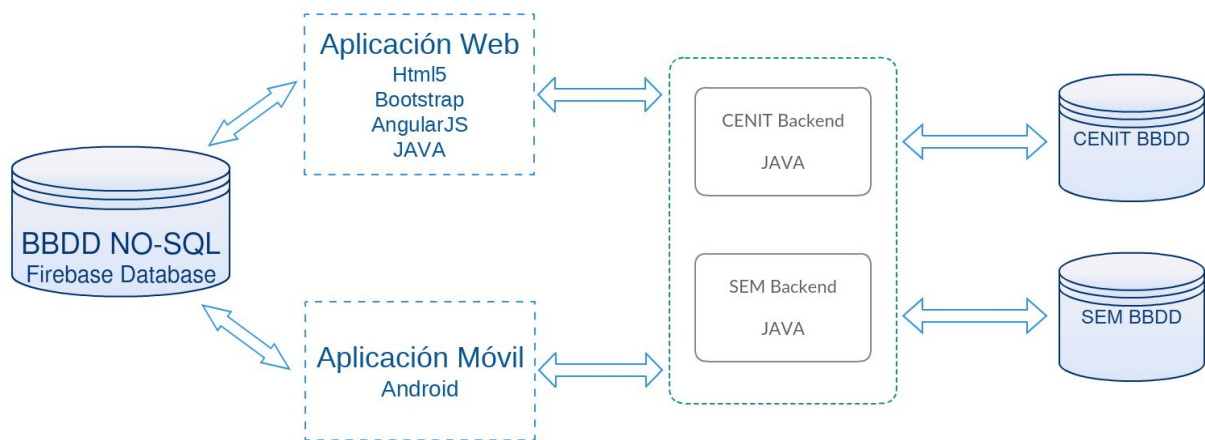


Figura 5.1. Arquitectura del proyecto Steer

Arquitectura alternativa

Una vez finalizada la aplicación, nos surgió una idea alternativa a la arquitectura realizada, que difiere un poco de la actual. Dependiendo de su aprovechamiento puede o no resultar beneficiosa, pero nos pareció interesante comentar esta variante, ya que como todo sistema, se puede elaborar de múltiples maneras.

La idea consiste en crear una capa entre, las aplicaciones web y mobile, y los sistemas donde se consumen los datos, es decir, los servicios brindados por el CeSPI. Al hacer esto tendremos un esquema modificado de la estructura original del proyecto. Esta variante nos indicaría que, las aplicaciones desarrolladas se comuniquen con esta nueva

capa, haciendo que la misma se comunice a la vez con los servicios REST de estacionamiento medido y todo lo que respecta a los eventos.

Esto nos provee un gran beneficio, que es el de la transparencia. Esto significa que desde las aplicaciones no debemos preocuparnos si algún servicio que efectúan las operaciones esenciales ya mencionadas cambian en algún momento, ya que siempre se comunicaran con la nueva capa y ésta será la encargada de resolver las llamadas REST a los servicios de estacionamiento y eventos, logrando así por ejemplo, el hecho de no tener que rearmar la aplicación móvil antes cambios en alguno de los backends, ya que siempre se tratará con esta capa. La siguiente imagen describe de manera gráfica lo explicado anteriormente.

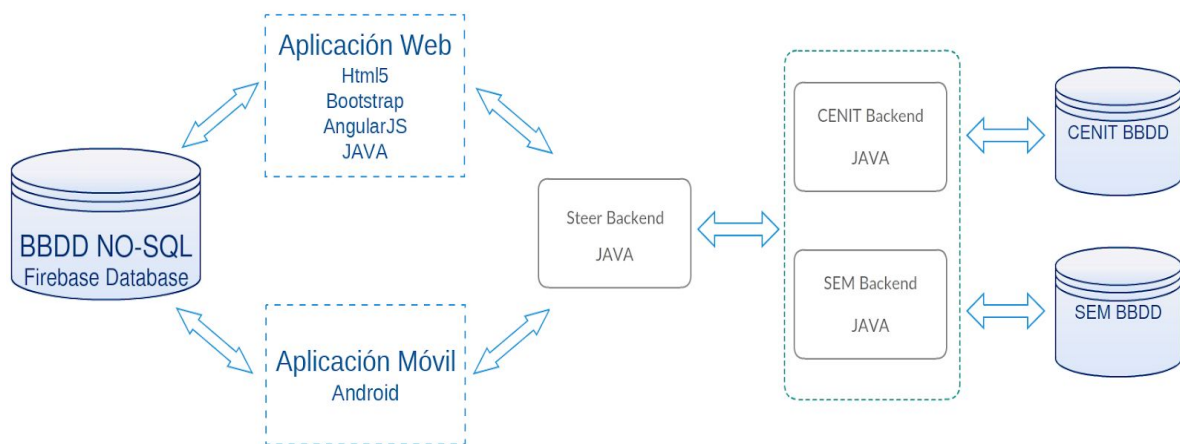


Figura 5.2. Arquitectura alternativa de Steer a la elegida en primera instancia

5.2.2 Buenas prácticas y preferencia de herramientas

Lo tomamos como un conjunto de reglas que tienen el fin de mejorar la calidad del software, brindando en el código ya sea del backend o del frontend legibilidad, facilidad de mantención, y soluciones simples a problemas recurrentes.

Se utiliza Android como plataforma de desarrollo por sus múltiples ventajas., se puede comenzar haciendo referencia a lo que se explicó anteriormente, el código de Android es abierto, esta libertad de código permite adaptar Android a bastantes dispositivos móviles, entre ellos, teléfonos inteligentes, tablets, relojes. Del mismo modo, al ser código abierto garantiza que, en caso de haber un bug o error, sea detectado con mayor presteza, al no tener ningún impedimento de acceso al código ni depender de nadie para el acceso al

mismo. Asimismo provee acceso a una gran cantidad de servicios, tales como GPS, bases de datos, reconocimiento y síntesis de voz, mapas, entre otros.

Cabe destacar que la aplicación Steer, fue escrita íntegramente en Inglés, por considerarlo una buena práctica debido a que:

- Es el idioma más difundido en el mundo de la informática, ya sea por historia misma de la computación como por nivel de hablantes del mismo.
- Evitamos problemas con palabras en el español por caracteres que suelen crear conflictos en líneas generales, tales como los acentos y la letra ñ.
- Mantenemos los mismos lineamientos que nos encontramos con el idioma del lenguaje trabajado y las APIs utilizadas, sustentando la idea de un código más entendible y prolijo a quién se lo dé a disposición.

5.2.3 Material design y las buenas prácticas de diseño

Para el diseño de la aplicación se usó material design, cuyo propósito es guiar el diseño visual, de movimientos y de interacción entre las diferentes plataformas. Entre los aspectos más interesantes que se encontraron resaltamos la simplicidad con la que se define visualmente la aplicación y el acceso a diferentes paneles, por lo que se logró una aplicación minimalista que minimiza la distracción al interactuar con la pantalla del sistema IVI del automóvil.

Para el desarrollo de Steer resultó útil el uso de material design en el menú lateral que contiene las configuraciones generales de la aplicación. Esto permite una buena experiencia de usuario, porque provee una navegación simple, directa y sin generar ningún tipo de distracción como podría hacerlo un menú con efecto al desplegarse o replegarse.

5.2.4 Persistencia lado cliente

Teníamos la intención de lograr el manejo de la infraestructura de manera simple, por lo cual optamos por Firebase como plataforma para esta tarea. La ventaja que encontramos es que nos brinda base de datos en tiempo real, sistemas de autenticación y almacenamiento estático. Además, la posibilidad de disponer de un SDK, de manera de poder integrarlo fácilmente a nuestra aplicación móvil, y un servicio REST, para la aplicación web.

Firebase almacena los datos en formato JSON⁸, es decir, que persiste los datos de manera no relacional. Esto resulta un beneficio, no es necesaria una herramienta de mapeo

⁸ Es un formato de texto ligero para el intercambio de datos.

objeto-relacional, el formato JSON puede ser serializados a objetos, y viceversa, tanto en Android como en PHP.

Después de analizadas las ventajas y desventajas de trabajar con bases de datos relacionales y no relacionales, y habiendo experimentado con ambas posibilidades, decidimos utilizar una base de datos no relacional y en particular, esta herramienta, que además posibilita diferentes optimizaciones y funcionalidades en comparación de una base de datos relacional. La API está diseñada para permitir sólo operaciones que puedan ejecutar rápidamente. Gracias a esto, la experiencia en tiempo real, que es parte del funcionamiento del proyecto se ve beneficiada sin afectar las capacidades de respuestas.

En síntesis, utilizar esta plataforma, mantenida por Google, simple de utilizar y con los alcances necesarios para los objetivos planteados, nos ha facilitado el desarrollo y nos parece una muy buena alternativa para la problemáticas con las que nos enfrentamos los desarrolladores de aplicaciones en tiempo real.

Capítulo 6

Steer, la aplicación

6.1 Aplicación móvil

Steer es una aplicación móvil que se integra a los sistemas IVI de los automóviles para facilitar el uso de las principales funcionalidades de los celulares mientras se conduce. La conexión es mediante usb, bluetooth o bien, vía wi-fi. De esta manera tendrá total acceso a las aplicaciones del dispositivo móvil compatibles con la tecnología del sistema IVI del automóvil, en nuestro caso MirrorLink, donde ésta lista incluye a Steer. Una vez hecha la conexión con el sistema, la pantalla del dispositivo móvil se bloquea por completo de manera de evitar, por seguridad, la interacción con el mismo mientras se está realizando la tarea de conducir. El usuario puede seleccionar la aplicación desde los controles disponibles en el automóvil, ya sea la pantalla touch o controles del volante.

El reto principal del sistema utilizado y la aplicación como fuente de consumo del primero, es el poder tener una aplicación diseñada para ser usada en los automóviles. Esto implica tener en cuenta un hecho tan esencial en este ambiente, como lo es, la distracción durante la conducción del automóvil.

La primera vez que se ejecute la aplicación se solicitará ingresar las credenciales de usuario, este proceso se realiza solo una vez ya que los datos quedarán almacenados en el dispositivo móvil. La Figura 6.1 muestra la pantalla de Login de la aplicación móvil.

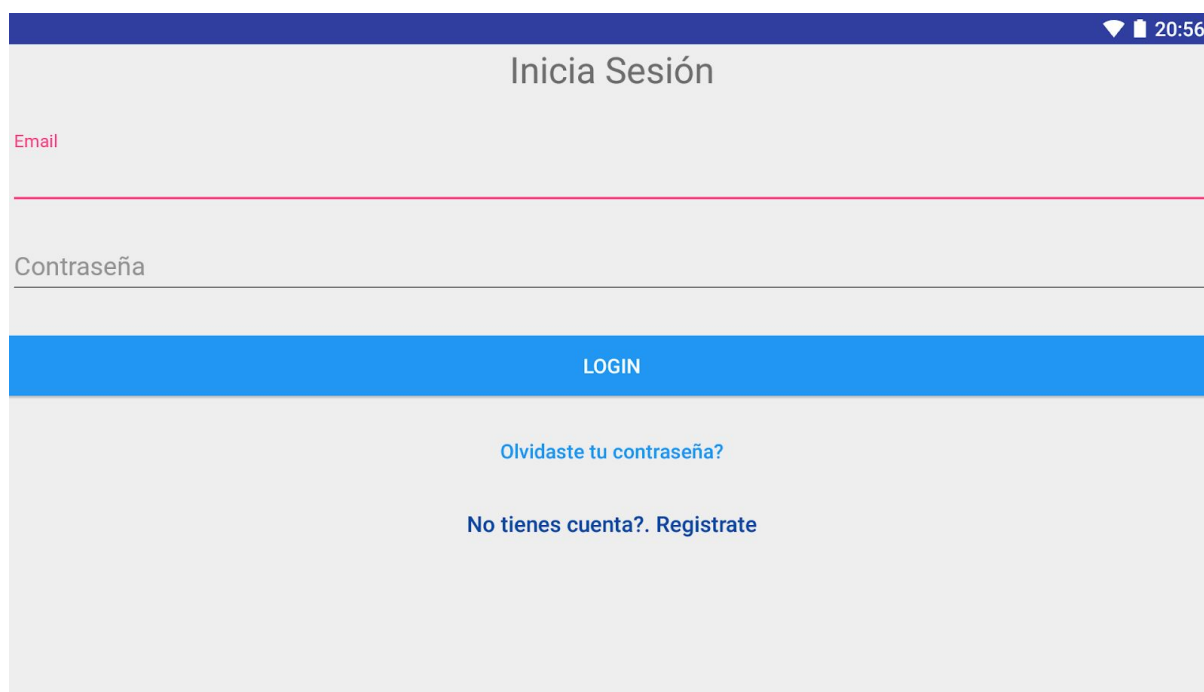
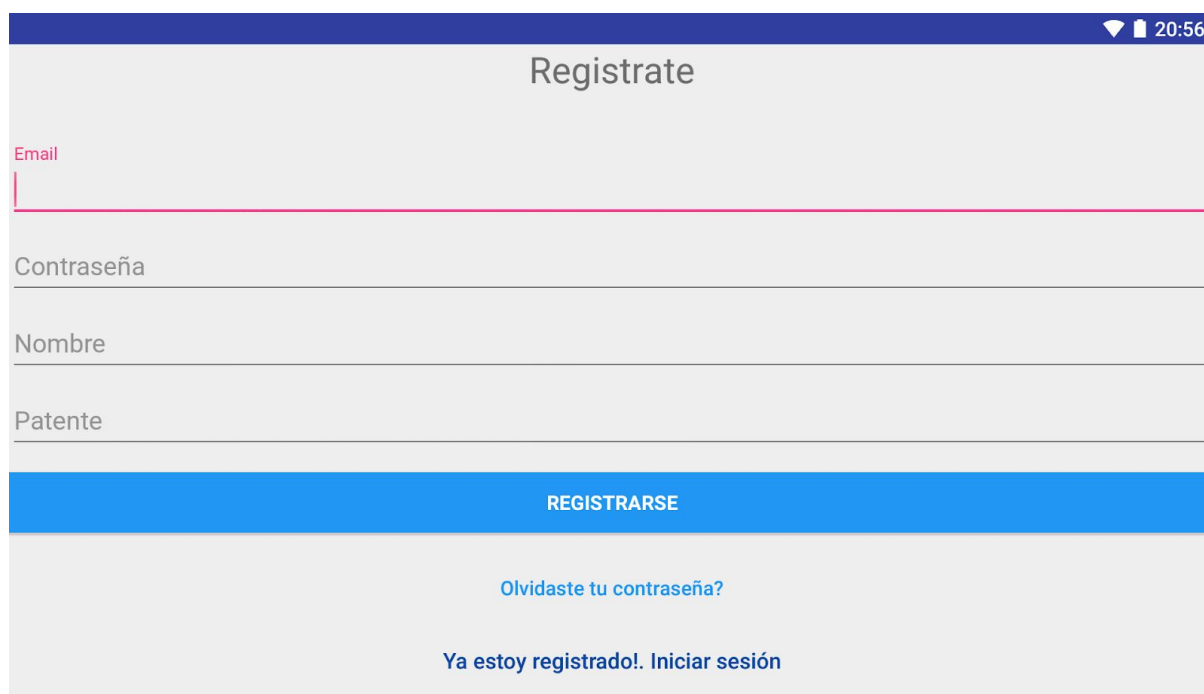


Figura 6.1. Pantalla de login APP Mobile de Steer.

En el caso de un usuario que aún no se ha registrado debe proceder a realizar el mismo ingresando datos personales básicos, nombre, apellido, correo, telefono y además la patente de su vehículo como muestra la Figura 6.2. Como la aplicación va a utilizar el servicio SEM de estacionamiento provisto por CENIT, para realizar transacciones de inicio y fin de estacionamiento, el usuario debe estar registrado en este sistema, en caso contrario, al realizar el registro en Steer, se realizará el registro en ambos sistemas. Si el usuario ya estaba dado de alta en SEM este paso se omite pues los datos necesarios para las transacciones son el teléfono y la patente, datos que se pueden obtener de la base de datos de Steer una vez dada el alta.



Registrate

Email

Contraseña

Nombre

Patente

REGISTRARSE

[Olvidaste tu contraseña?](#)

[Ya estoy registrado!. Iniciar sesión](#)

Figura 6.2. Pantalla de registro APP Mobile de Steer.

El proceso de registro o login se realizará solo una vez, los datos son guardados en el dispositivo móvil, y luego, cada vez que ingrese a Steer, el usuario tendrá acceso a todas las funcionalidades que provee la aplicación y podrá ver directamente el mapa con los servicios a su disposición como muestra la imagen de la Figura 6.3.

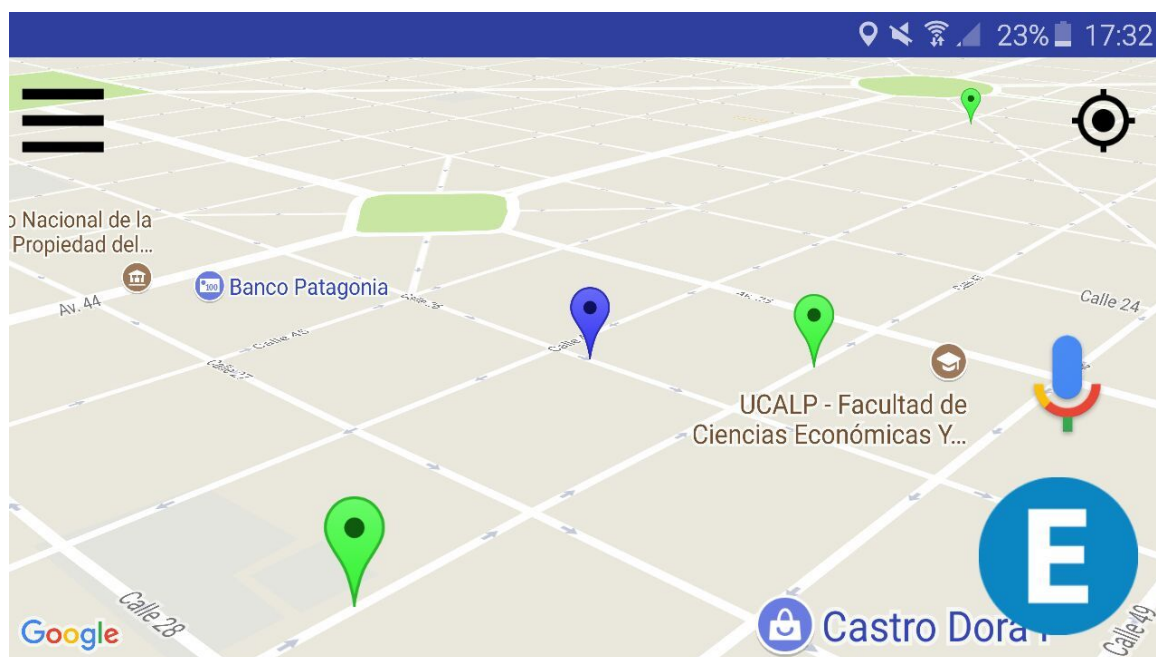


Figura 6.3. Captura de pantalla de la APP Mobile de Steer.

La pantalla central de la aplicación consiste de un mapa de navegación en el cual veremos la posición actual de nuestro vehículo, que se irá actualizando cada pocos segundos para realizar el seguimiento de nuestro vehículo. Además cuenta con una serie de botones simples del lado derecho de la pantalla, y menú desplegable, dónde se encuentra la configuración básica de la aplicación. Los botones son tres: el primero comenzando desde arriba es para centrar la cámara en la posición del usuario, debajo está el micrófono que capta una serie de comandos de voz predefinidos, para que el usuario interactúe con el sistema sin interacción con las manos y finalmente un botón con una “E” para iniciar o finalizar el estacionamiento. Estos detalles se especificarán más adelante para conocer el funcionamiento exacto de estos botones. Además existe un ícono que apunta la ubicación dónde se encuentra el conductor en tiempo real.

La importancia del sistema radica en proveer información esencial de estacionamiento medido del lugar, para lo cual, la aplicación va a tener diferentes eventos con los que se podrá interactuar.

El primero de ellos será cuando el automóvil ingresa a una zona de estacionamiento mientras está conduciendo. A partir de este momento ocurrirán una serie de sucesos.

- ❖ Se notificará mediante un mensaje de voz y un mensaje textual muy breve en la pantalla del sistema IVI, que ha ingresado a una zona en la que funciona el sistema de estacionamiento medido, y los datos del mismo como horarios y precio. De esta manera el usuario adquiere noción de esta información al momento de querer estacionar con el automóvil.

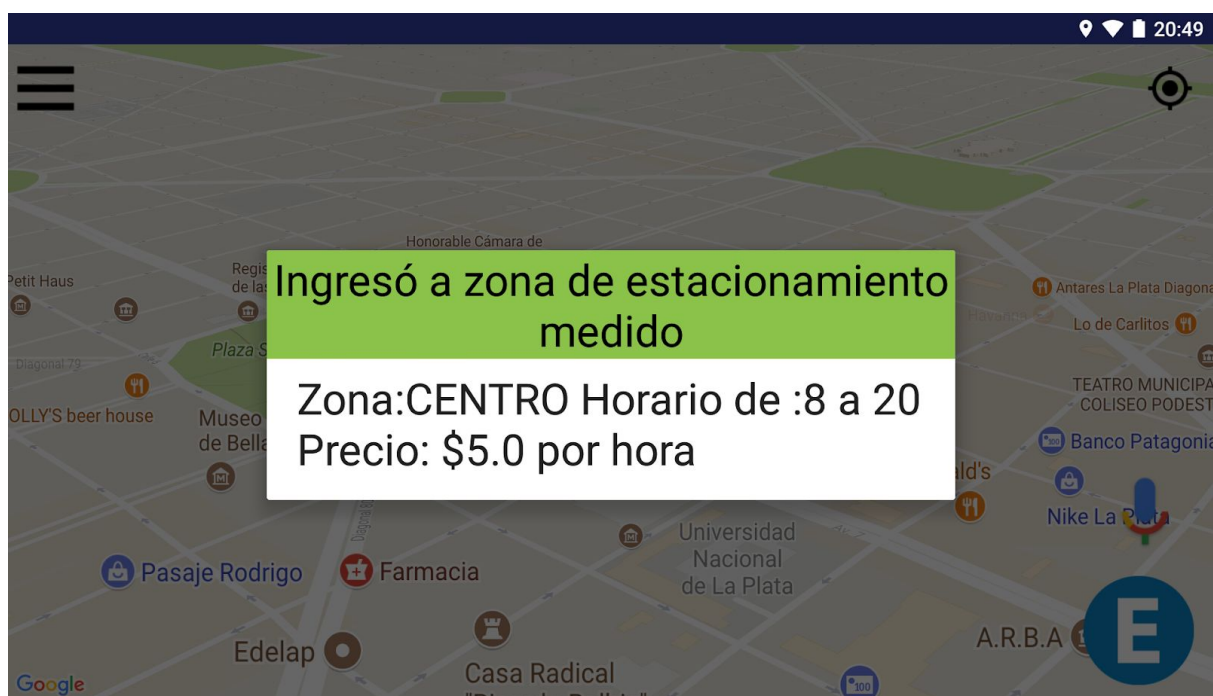


Figura 6.4. App Steer al ingresar a una zona de estacionamiento medido.

- ❖ El botón de iniciar estacionamiento, puede pasar por diferentes estados que serán identificados por el color del mismo. Inicialmente, se encontrara en color gris, que nos indica que no estamos en zona de estacionamiento y por lo tanto estará deshabilitado. Luego, al entrar en una zona de estacionamiento, este botón pasa a estar en un estado habilitado, esto quiere decir que visualmente el usuario verá ahora el botón de color azul, lo que le indica que el mismo está disponible para utilizarlo. Finalmente si ya ha iniciado el estacionamiento, el botón en color verde. Esto lo podremos apreciar en la Figura 6.3, que muestra de manera clara lo explicado previamente.

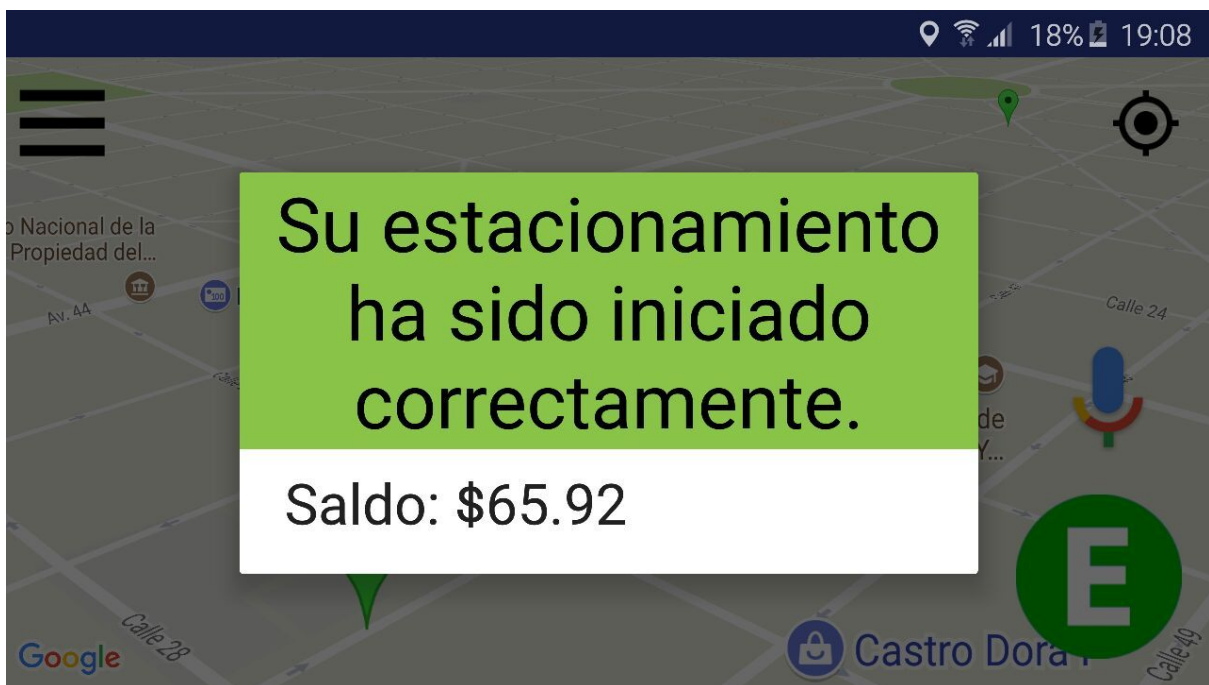


Figura 6.5. App Steer cuando se inicia correctamente un estacionamiento.

Por otro lado tenemos acciones que responden cuando el automóvil desea iniciar estacionamiento. Esta operación es una de las más interesantes que provee la aplicación, ya que las puede hacerse por diferentes vías. A partir de querer iniciar esta acción, tenemos las siguientes consecuencias.

- ❖ Una posibilidad para iniciar estacionamiento, es presionando el botón que representa esta operación dedicada, ubicada en la pantalla inferior derecha, con la letra “E”, la cual anteriormente dijimos, que cuando no está en zona de estacionamiento se encuentra en un estado deshabilitado, en color gris, y, si se

encuentra en zona operable cambia de estado a habilitado y color azul. Una vez presionado este botón, podemos obtener diferentes resultados.

- Correcto: Este resultado es el más esperado, sucede si, el usuario tiene crédito en su cuenta para poder costear el saldo mínimo necesario para iniciar estacionamiento en el sistema. Si esto ocurre, el usuario recibirá un mensaje de voz y otro en la pantalla del sistema IVI, comunicando al usuario que la operación fue exitosa, y, a partir de este momento el automovilista ya posee iniciado estacionamiento de manera correcta y en regla. También luego de esta operación, el estado del botón que acabamos de presionar cambia, pasando a verse en la pantalla de color verde, de manera de comunicar al usuario que tiene estacionamiento activo.
- Incorrecto: Este es otro posible resultado luego de aplicar el evento mencionado. Sucede cuando el usuario no posee crédito suficiente para costear el saldo mínimo necesario para iniciar estacionamiento. Por lo tanto, luego de intentar aplicar la operación, se devuelve un resultado tanto por mensaje de voz, como por mensaje en la pantalla del sistema del tablero del automóvil, con la respuesta de que no posee saldo suficiente para realizar la operatoria. El botón presionado permanecerá en el mismo estado en el que se encuentra, es decir, habilitado y en color azul.
- Otros: Existen otros tipos de mensajes cuando el usuario quiere iniciar estacionamiento pero que no pertenece a ninguno de los anteriores. Existen casos en los que el día no requiere estacionamiento, por ejemplo un día feriado, por lo tanto al querer iniciar la operatoria en cuestión, se devuelve una respuesta con el mensaje por el que no es necesario iniciar.

Otra operación fundamental que existe, es la de la finalización del estacionamiento. Una vez que el automóvil deja su lugar de estacionamiento, para retirarse del lugar, en la aplicación debe finalizarse el inicio realizado a través de alguna metodología existente. La más ágil y simple es utilizando el mismo botón que se usó anteriormente, dedicado para esta acción específica sin mayores complejidades. El funcionamiento de este es de la siguiente manera:

- Como se mencionó anteriormente, cuando el el usuario generó el evento de iniciar estacionamiento, el botón se encontrará en color verde, lo cual indica, que el anterior evento se realizó de manera correcta. Al presionar el mismo, el sistema notificará mediante mensaje de voz y de texto en la pantalla del sistema IVI, la finalización

correcta del estacionamiento. Luego de esto, el botón cambiará a su estado inicial y el color será nuevamente azul, lo que nos indicará que ya realizamos el corte de manera adecuada.

- Otra metodología existente para que ocurra esta acción es de manera automática. Esto sucede cuando el usuario se retira con el automóvil olvidándose de finalizar estacionamiento. Al salir de la zona de estacionamiento, se finaliza de manera automática, entendiendo que ya se está retirando del lugar y no usará más el sistema. Esta alternativa surge de la necesidad de ayudar al conductor con las bondades que tenemos gracias a la geolocalización y a la información que se maneja en la aplicación.

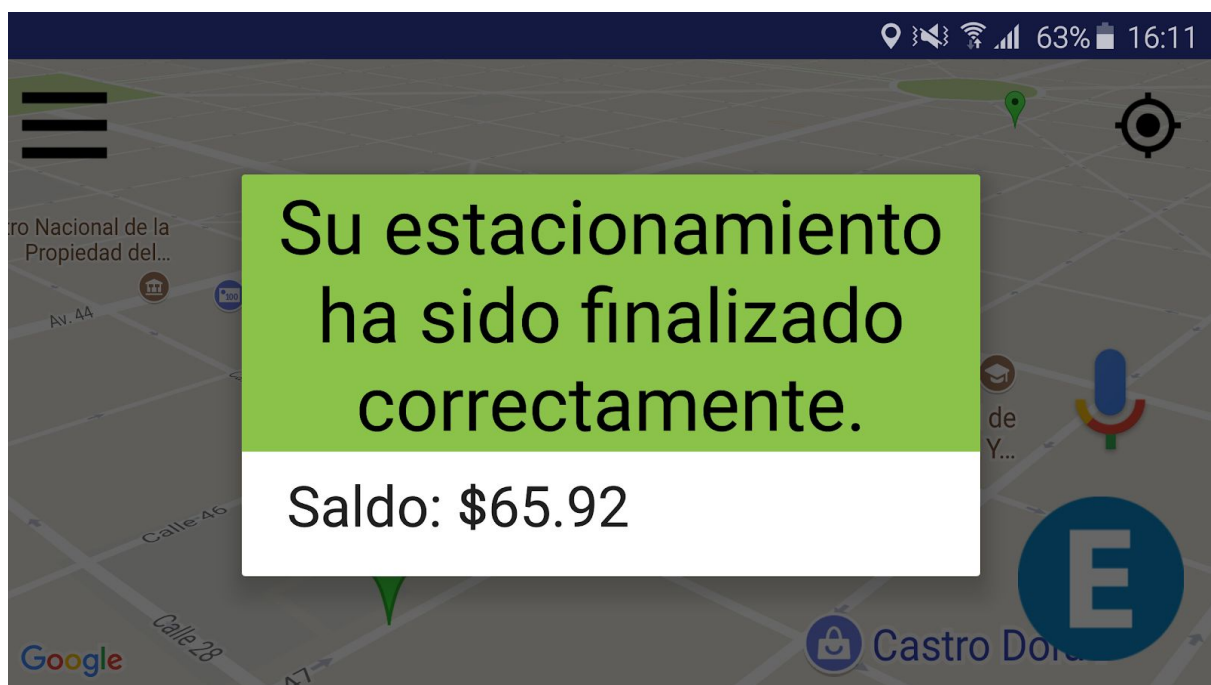


Figura 6.6. Comportamiento visual de la aplicación cuando se finaliza estacionamiento.

El otro servicio que provee Steer, es el de generar diferentes tipos de eventos mientras se está realizando la tarea de conducir. De este modo, estamos aportando información al sistema de algún evento, que será enviado para moderar el mismo y, esto si es aprobado, nos aparecerá en el mapa de nuestra aplicación. En consecuencia, también tendremos información acerca de las rutas que estamos manejando, y esta la podremos utilizar para lo que más nos beneficie.

Por ende, pasaremos a detallar el otro aspecto técnico de la aplicación. Esto nos lleva al botón con el micrófono mencionado al principio del capítulo, que responde a comandos de voz. Esta funcionalidad hace muy flexible a la aplicación, ya que son comandos personalizados, y además, tiene la posibilidad de añadir tantos como se deseen adicionales a los ya existentes. También, los comandos de voz, mientras se está realizando la tarea de conducir, son los apropiados para interactuar con un sistema externo ya que minimiza la distracción. A continuación se va a explicar otra de las partes interesantes de Steer, que es responder a eventos de voz del usuario y determinar una acción a seguir. Para realizar esto, se presiona el botón del micrófono y se dice alguna de las tantas palabras claves definidas.

Por un lado puede responder a una serie de comandos que se relacionan con la parte de estacionamiento medido.

- **Iniciar, estacionar, iniciar estacionamiento:** Son palabras claves que el sistema interpreta como maneras de invocar al evento que inicia estacionamiento medido. Las respuestas ante este llamado, son las mismas que al utilizar el botón (y estamos en zona de estacionamiento medido). Pero además de esto tenemos otro resultado. Este sucede cuando el estacionamiento ya está iniciado en la aplicación y, se llama a través de esta metodología nuevamente a iniciar. Ante esta situación, se responde por medio de un mensaje de voz y otro en la pantalla del sistema, que ya tiene estacionamiento activo.
- **Fin, finalizar, finalizar estacionamiento:** Son palabras claves que el sistema interpreta como métodos para invocar la finalización del estacionamiento medido. En este caso las respuestas como en el caso anterior, incluye a las respuestas dadas anteriormente. Asimismo, tenemos otro tipo de respuesta, que será, cuando se invoca a esta operación sin tener estacionamiento en curso. Por tanto, la aplicación va a devolver un mensaje informando esto, a través de voz y por la pantalla del sistema IVI.
- **Consultar, saldo:** Estas palabras claves son un extra, para que el usuario tenga la posibilidad de conocer su estado en el sistema respecto a su cuenta. Simplemente informará al mismo la cantidad de dinero que posee en su cuenta corriente. Al ser

una función meramente informativa, el botón principal “E”, que realiza este tipo de operaciones y, explicadas antes, no incluye nada con respecto a esta operación.

Inicios de EM por agente													
Agente	0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60
Android	0	1	0	0	0	0	0	0	0	0	0	0	0
Facebook	0	0	0	0	0	0	0	0	0	0	0	0	0
iPhone	0	0	0	0	0	0	0	0	0	0	0	0	0
Sur	0	0	0	0	0	0	0	0	0	0	0	0	0
Telegram	0	0	0	0	0	0	0	0	0	0	0	0	0
Web	0	0	0	0	0	0	0	0	0	0	0	0	0

Estacionamientos	Total	Android	Facebook	iPhone	Sur	Telegram	Web
Activos actuales	0	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)
Total diario acumulado	1	1 (100,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)	0 (0,00 %)

Estacionamientos iniciados			
Estacionamientos de hace		5 minutos	
Fecha y hora de inicio	Fecha y hora de fin	Patente	Agente de Inicio
04/03/2018 16:11	04/03/2018 16:11	ZIO989	Android
<< < 1 / 1 > >>			

Desarrollado por Ce.S.P.I. U.N.L.P. - Municipalidad de Torti - Todos los derechos reservados - Última versión v2.04 © 2014 - 2018

Figura 6.7. Recorte del sitio web del CeSPI, que indica que el estacionamiento se inició y finalizó correctamente a través del agente de Android.

Por otro lado, las acciones posibles a realizar a través de los comandos de voz, son los de generar alertas. Son dos los tipos de alertas los que se pueden producir desde la aplicación y ambos responder a instrucciones predeterminadas, como lo son los explicados anteriormente. Se detalla cada una de las órdenes a la cual se responde para generar alguna de las alertas cuando se presiona el botón del micrófono.

- ❖ **Bache:** Este comando se utiliza cuando el usuario percibe un hundimiento de la ruta en la que maneja, y quiere alertar en el sistema acerca de la existencia del mismo. La aplicación cuando recibe esta instrucción, toma las coordenadas del lugar donde se produce el pedido, a través de las herramientas utilizadas dentro de la aplicación, se obtiene la calle y altura del lugar y envía entre otro tipo de información, la alerta hacia el sistema. Si se procesó bien la solicitud, se notifica al usuario por medio de un mensaje de voz, que la petición se realizó de manera correcta. El resultado final del pedido se explicará de manera global luego de detallar cada alerta.
- ❖ **Manifestación:** Este comando es utilizado cuando el usuario se topa una marcha multitudinaria en el área que está circulando, y quiere advertir este evento a otros usuarios de la aplicación. Cuando se notifica este tipo de evento, al igual que el

anterior caso, se toman los datos del punto geolocalizado y se envían al sistema a través de la aplicación, para procesar estos datos.

De manera genérica procederemos a explicar qué sucede con los datos una vez enviados desde la aplicación. Sin interesar qué tipo de evento se esté solicitando, la información recolectada se enviará a moderar. A un usuario moderador le llegará un mensaje con los datos del evento en cuestión, y verificará por sus medios si es correcto este testimonio, de manera de tener la potestad de rechazar la solicitud o bien aprobar la misma. Siguiendo el caso negativo, es decir, rechazar la petición, el ciclo de la alerta finaliza allí. En el caso que sea positivo el resultado de la solicitud, el mapa de la aplicación de los usuarios se actualizará con un ícono representativo asociado al evento, teniendo la información en tiempo real de lo que está sucediendo en las rutas que circula.

Dado que puede haber un gran número de eventos enviados, y la vista del mapa puede volverse engorrosa y por ende, confundir al usuario, éste sólo verá en su pantalla la cantidad que se encuentran en un radio determinado teniendo como centro su localización actual. De este modo, el usuario puede alertar hechos y al mismo tiempo mantenerse informado por eventos realizados por él o por otros usuarios de la aplicación.

El menú de Steer se accede desde la vista principal mediante el uso del botón que se encuentra arriba a la izquierda de la misma, el cual despliega el menú principal de la aplicación, que consta de la información básica al usuario, funciones y, además una serie de ítems de configuración de la aplicación tal como se ve en la Figura 6.8.

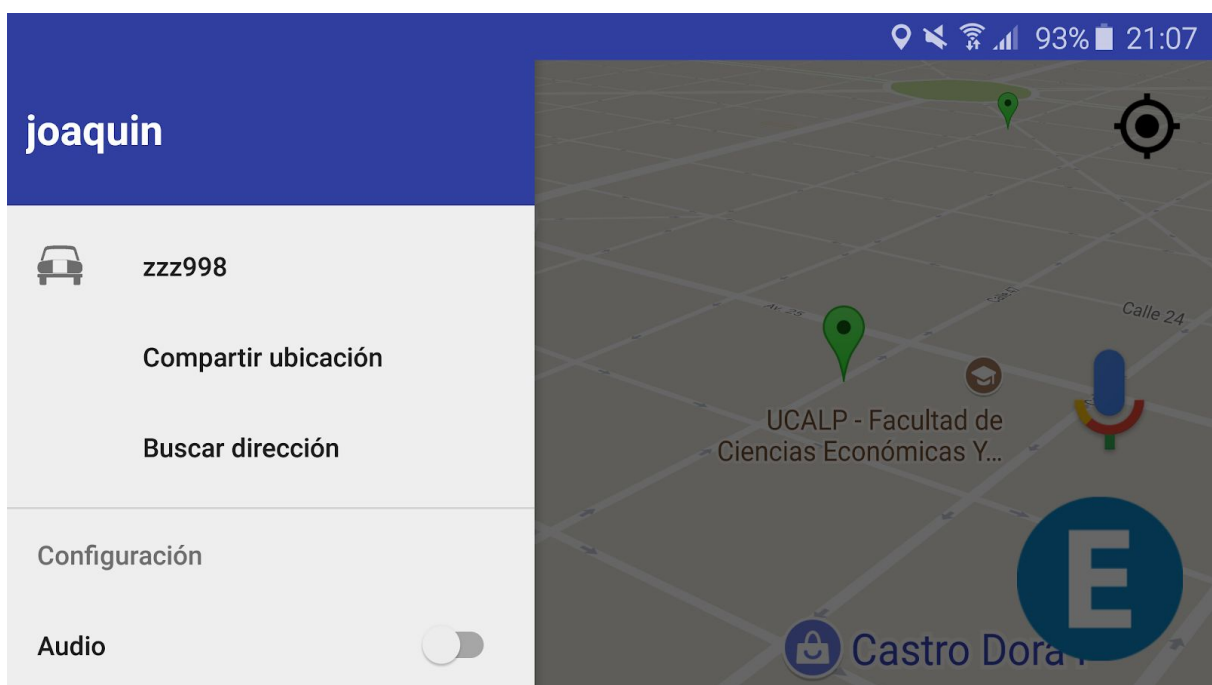


Figura 6.8. Pantalla con el menú principal de Steer.

La parte superior del menú fue pensada de manera tal de tener una identificación de la aplicación por medio del nombre de la misma y además añadir el nombre de usuario del conductor. Además se pensó que, agregar íconos al lado de textos del menú, haría más intuitivo el uso del éste, ya que se recordaría más fácil cada ítem.

Comenzando de arriba hacia abajo de la Figura 6.8, el primero muestra un ícono de una automóvil sencillo con el texto del número de patente del usuario conductor, este entonces será, la patente usada en el momento que se estacione el automóvil o finalice estacionamiento, o que requiera esta información para una acción en particular.

- **Compartir ubicación**

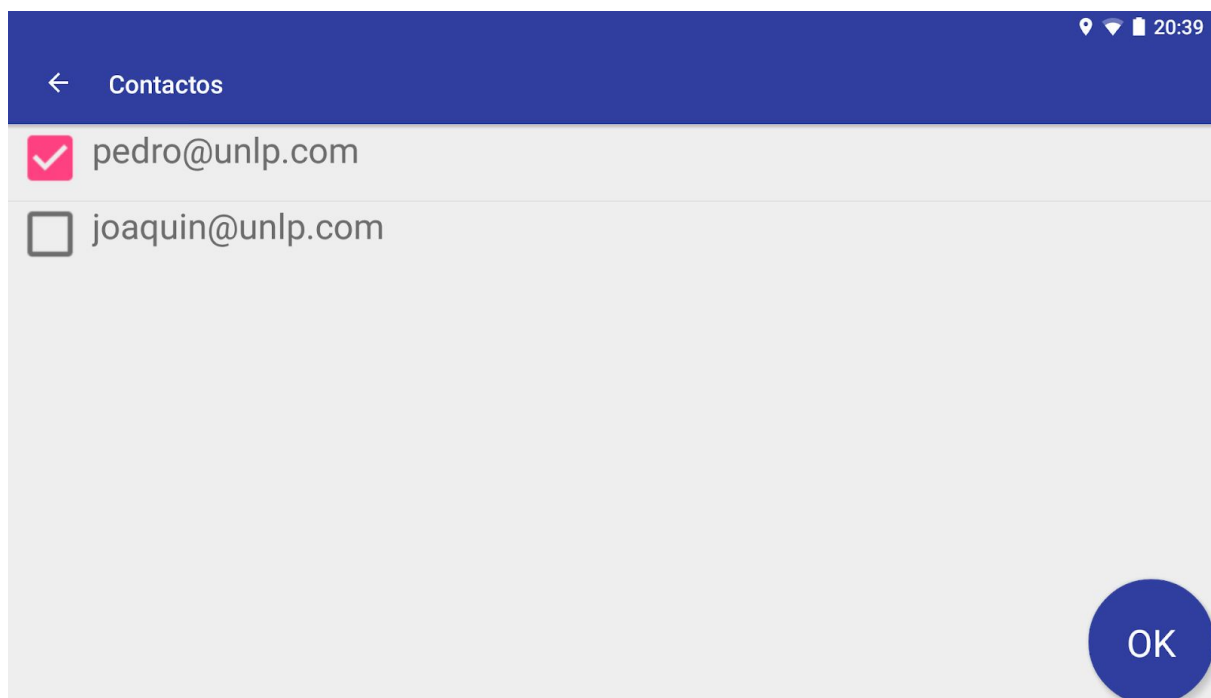


Figura 6.9. Pantalla de selección de contactos para compartir ubicación.

Otra funcionalidad que tendrán disponible los usuarios de Steer será la posibilidad de conceder permisos, a otros usuarios, para que visualicen su ubicación, en el mapa, en tiempo real. De este modo se puede realizar un rastreo o seguimiento, mediante el

consentimiento del usuario que puede ser útil en caso donde usuarios de distintos vehículos se dirijan a un mismo destino, o en caso donde se desea saber la posición de un conocido por cuestiones de seguridad o coordinación de encuentros.

Los usuarios pueden encontrar en el menú desplegable la opción de compartir ubicación, al seleccionarla se abrirá una nueva ventana de la aplicación con la lista de contactos con un checkbox para cada y al marcarlo estaremos dando el consentimiento para que puedan ver su ubicación durante, esta marca se puede quitar en cualquier momento o automáticamente en un lapso de 24 horas.

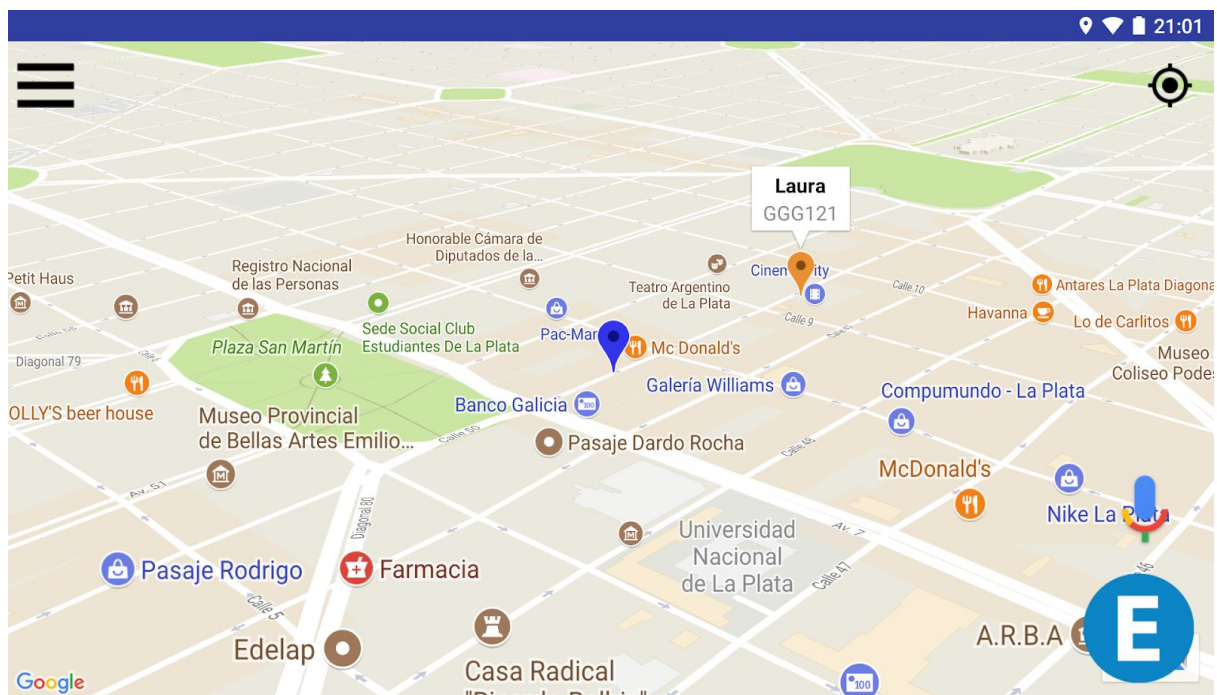


Figura 6.10. En el mapa podemos ver la posición del usuario que nos envió su ubicación

Los usuarios podrán agregar nuevos contactos desde la aplicación Web, esto será explicado posteriormente.

Por otro lado los usuarios a los que se le compartió la ubicación, de forma automática, verán en el mapa un nuevo marcador que representa nuestra ubicación. Si el usuario lo desea podrá deshabilitar esta funcionalidad (de manera que solo verá su propia ubicación en el mapa) mediante una opción de configuración en el menú desplegable.

Al igual que en la función de buscar direcciones, por cuestiones de seguridad, se debe evitar generar distracciones mientras se conduce, es por eso que la opción de compartir ubicación también está habilitada solamente cuando el vehículo no se encuentre en movimiento.

- **Buscar dirección**

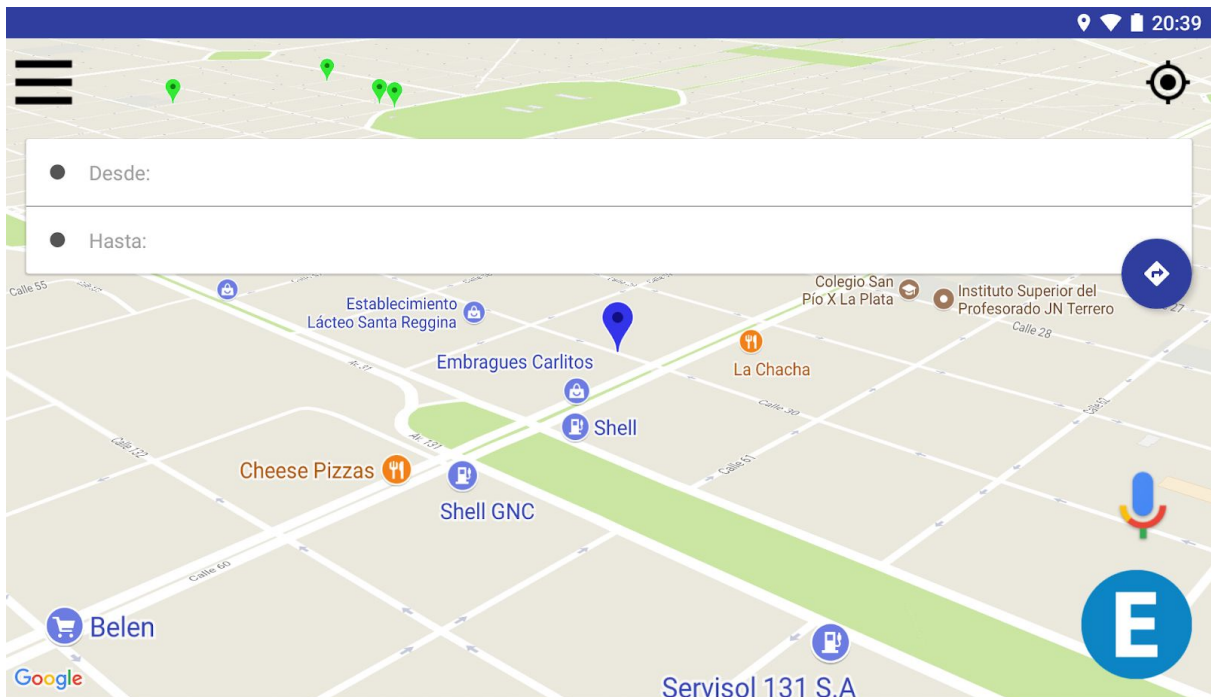


Figura 6.11. En el mapa los controles de búsqueda de direcciones

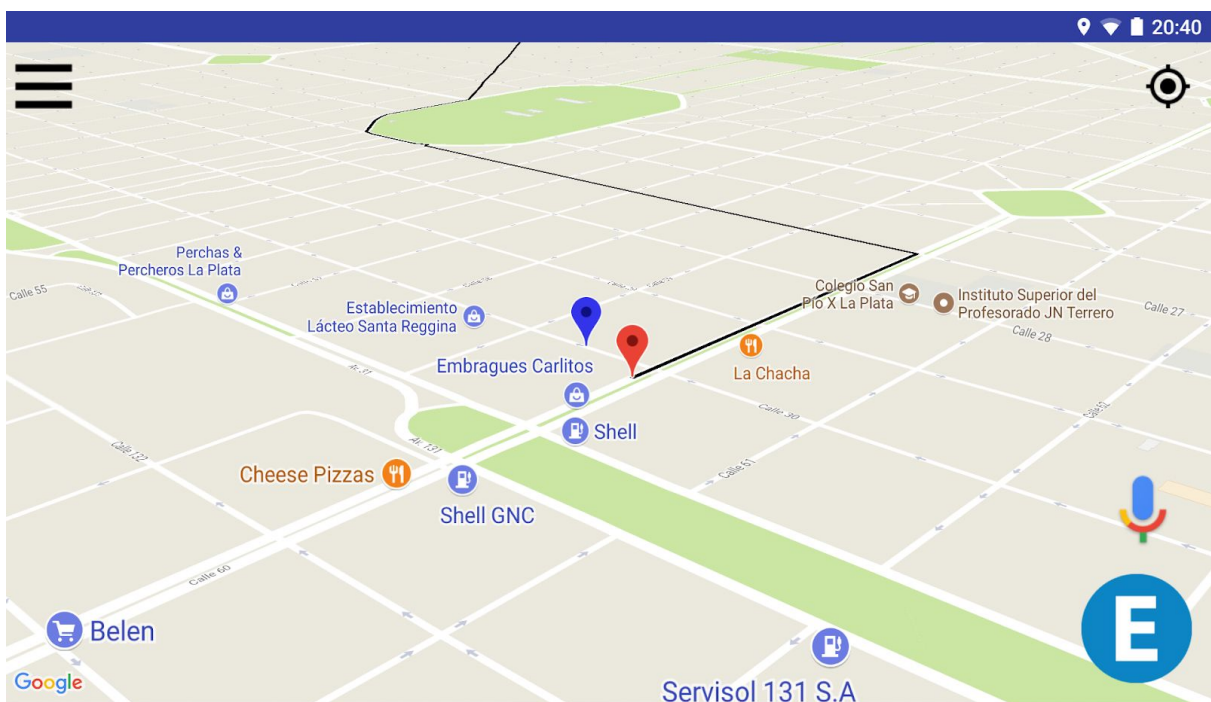


Figura 6.12. En el mapa podemos ver la ruta entre las direcciones ingresadas

Al ser una aplicación de navegación, nos pareció necesario proporcionar la funcionalidad para que el usuario pueda buscar una dirección. Al hacer esto el sistema le devolverá una ruta que enlaza la posición actual del usuario con el destino, la misma será dibujada en el mapa. Para este propósito se utilizará Google Maps Directions API, servicio que ya fue detallado anteriormente.

Accediendo al menú desplegable el usuario tendrá la opción de buscar un destino, al seleccionarla se desplegará el teclado para ingresar la dirección manualmente. Conscientes de que este accionar es causante de distracciones, esta función sólo será permitida si el vehículo no está en movimiento, la API de MirrorLink dispone de la capacidad de detectar este suceso, por lo que vemos provechoso el uso de la misma para estos casos particulares en los cuales ponemos ante todo la regla de la menor distracción posible del usuario conductor.

- **Audio**

Luego tenemos la parte del menú de “configuraciones” el ítem de “audio”. Vemos importante tener esta opción en la configuración de la aplicación, ya que la aplicación móvil cuenta con sonido en los comandos de voz en las mayorías de las operaciones que se pueden realizar. De modo tal que si el usuario cree que es muy invasivo el uso del sonido en cada instancia de éstas descritas, tiene la posibilidad de deshabilitar o habilitar nuevamente el audio del mismo.

- **Detener seguimiento**

Esta opción permite definir si aceptamos las peticiones de seguimiento de otros usuarios de Steer.

6.2 Aplicación Web

Se vio la necesidad de tener toda la información del usuario que ha registrado históricamente desde la aplicación móvil, por lo tanto se desarrolló una aplicación accesible desde cualquier navegador web, de manera tal de consultar diferentes tipos de datos del

usuario y de la información disponible de la aplicación móvil que se puedan ver a través de este sistema.

La aplicación consiste en un sitio web que interactúa con un backend de manera de obtener diferentes tipos de datos para renderizar en la pantalla principal del navegador. Cuando se ingresa al sitio, se muestra una página donde se le solicita al usuario ingresar las credenciales para acceder al sistema (*Figura 6.13*), las mismas serán una combinación de un nombre de usuario, que usualmente será el número de teléfono de la persona que posea instalada la aplicación Steer en su dispositivo móvil, y además se necesitará una contraseña, que será la misma con la que el usuario ingresará a la aplicación móvil. Una vez ingresado el usuario y la contraseña de su usuario Steer, los mismos son enviados a la base de datos, para validar que los mismos sean correctos, si esto es verdadero, se mostrará en la pantalla la pantalla principal del sitio web de Steer.

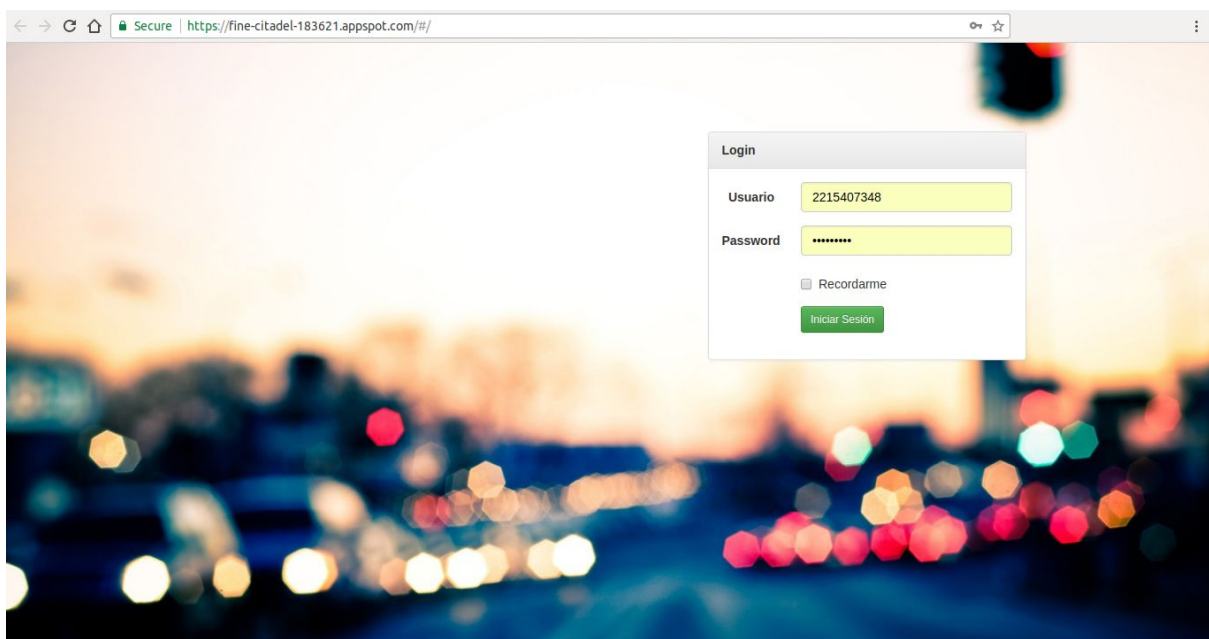


Figura 6.13. Pantalla de login del sitio web de Steer.

Una vez que el usuario ingresa al sitio correctamente, se renderiza en la pantalla principal el home del mismo, (ver *Figura 6.14*). A continuación se describe las diferentes operaciones posibles que nos proporciona el sitio web de Steer, para que sirva cada menú, y los resultados que se obtiene al utilizar cada uno de ellos.

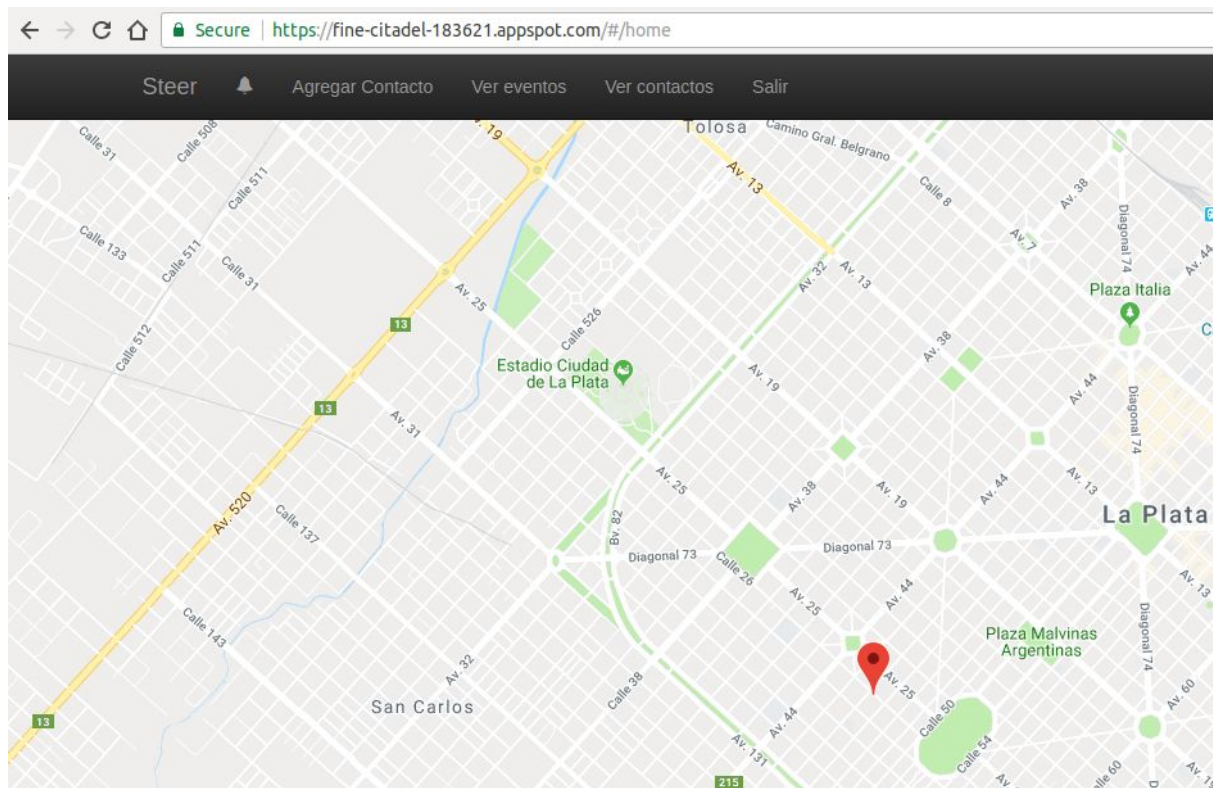


Figura 6.14. Imágen del sitio web Steer inicial.

Como se ve en la imagen, hay un menú en la parte superior, y un mapa donde se mostrará diferente tipo de información que se quiera consultar. La barra de navegación consisten en una serie de menús, los cuales de manera independiente, cada uno de ellos cumple con un objetivo específico, seguidamente, se explicarán estos de manera detallada.

- Ver eventos

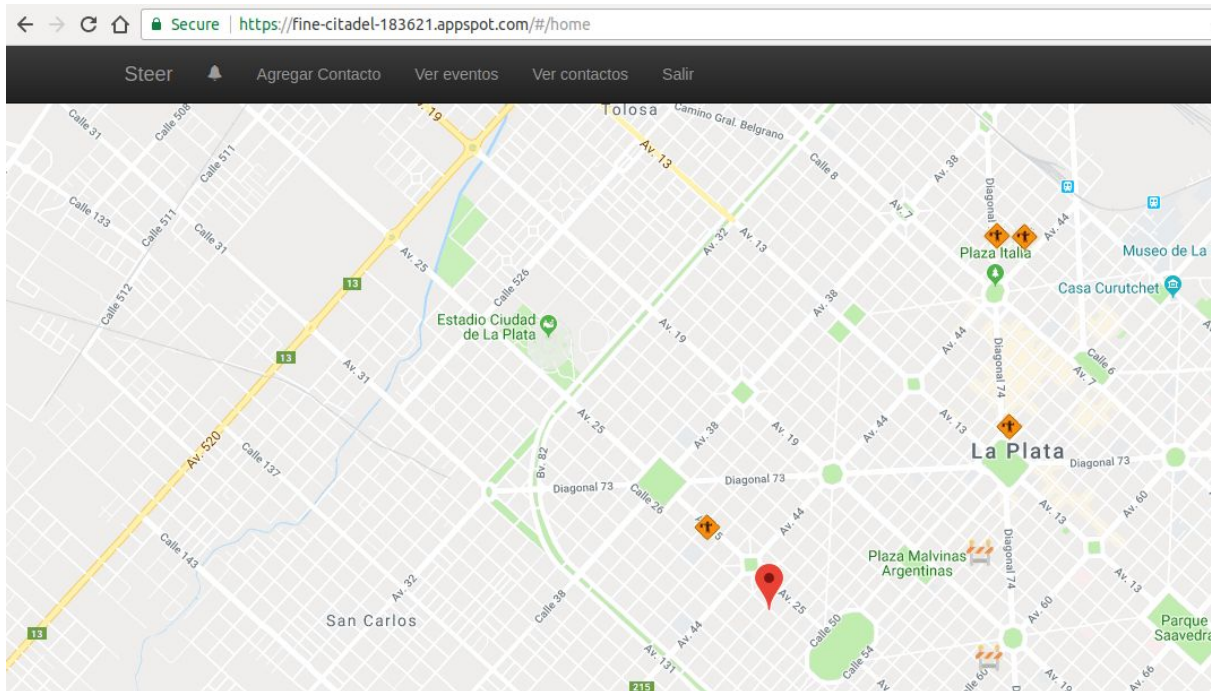


Figura 6.15. Mapa con eventos reportados por usuarios.

Esta función consulta, a través de los servicios otorgados por cenit, los eventos reportados por los diferentes usuarios. Esto lo hace completando el mapa, un ícono con una breve descripción del evento en cuestión en cada coordenada que haya un evento. Esto le permitirá al usuario revisar si sus eventos reportados fueron aprobados y además poder ver otras alertas aprobadas por otros usuarios, para así conocer el estado de los eventos y tener el conocimiento de posibles altercados que pueda tener durante su conducción momento antes de trasladarse por la urbe.

- Agregar contacto

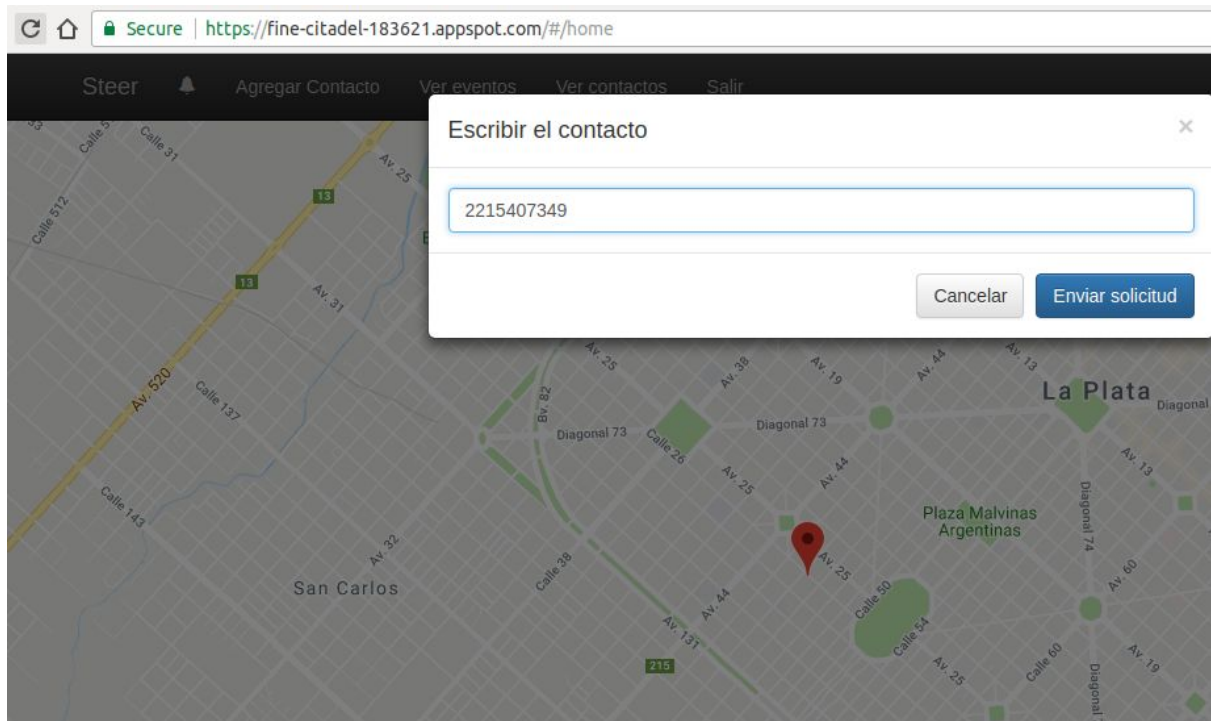


Figura 6.16. Acción de agregar un contacto

El presente menú tiene como objetivo primordial enviar una solicitud de contacto a un usuario. La consecuencia de esta acción es poder mantener una conexión con otra persona, de manera tal, de poder ver diferente tipo de información que pueden ser compartidas entre ellos, por ejemplo, la localización del otro contacto en tiempo real para poder seguir la ubicación de un modo cercano. La acción principal de este menú puede tener dos resultados posibles de esta petición a partir de lo que decida el receptor, tal como se explicó en el segmento anterior. Ante el caso negativo, el solicitante podrá enviar una nueva solicitud de contacto al mismo usuario en otra oportunidad. En el caso que la respuesta sea positiva, nos añadirá en nuestra lista de contactos al usuario en cuestión y es lo que nos posibilita acceder a los beneficios de la funcionalidad descrita recientemente.

- Notificaciones

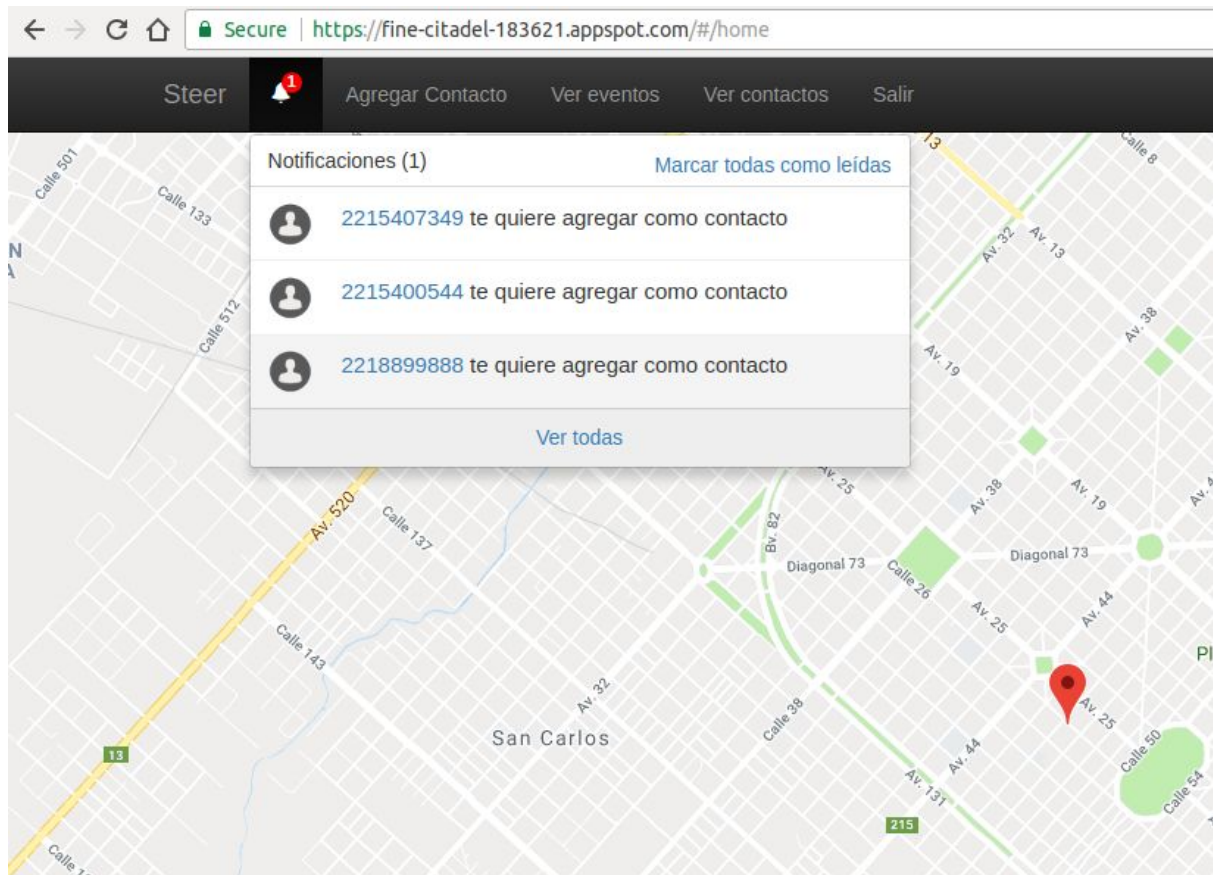


Figura 6.17. Menú de notificaciones de Steer.

Tal como podemos apreciar en la figura, este menú consiste en la imagen de una campana, haciendo reseña a un menú donde podemos encontrar notificaciones que son creadas por eventos, tal como puede ser el envío de una solicitud de contacto. Como se explicó anteriormente, cuando un usuario envía esta solicitud a otro usuario, al usuario receptor le aparecerá automáticamente una nueva notificación, con un globo que hace referencia a que tiene un nuevo aviso en su cuenta. A partir de allí, este tiene diferentes caminos posibles a seguir.

El primero de ellos es seguir la acción del aviso que nos abrirá una ventana modal, la cual es explicada en el subítem “*Ver Contactos*”, decidiendo allí el ciclo final de dicha solicitud. Por otro lado se puede a todas las notificaciones “*marcar como leídas*” a todas ellas, de modo tal de decidir luego qué hacer con dichas solicitudes.

- Ver contactos

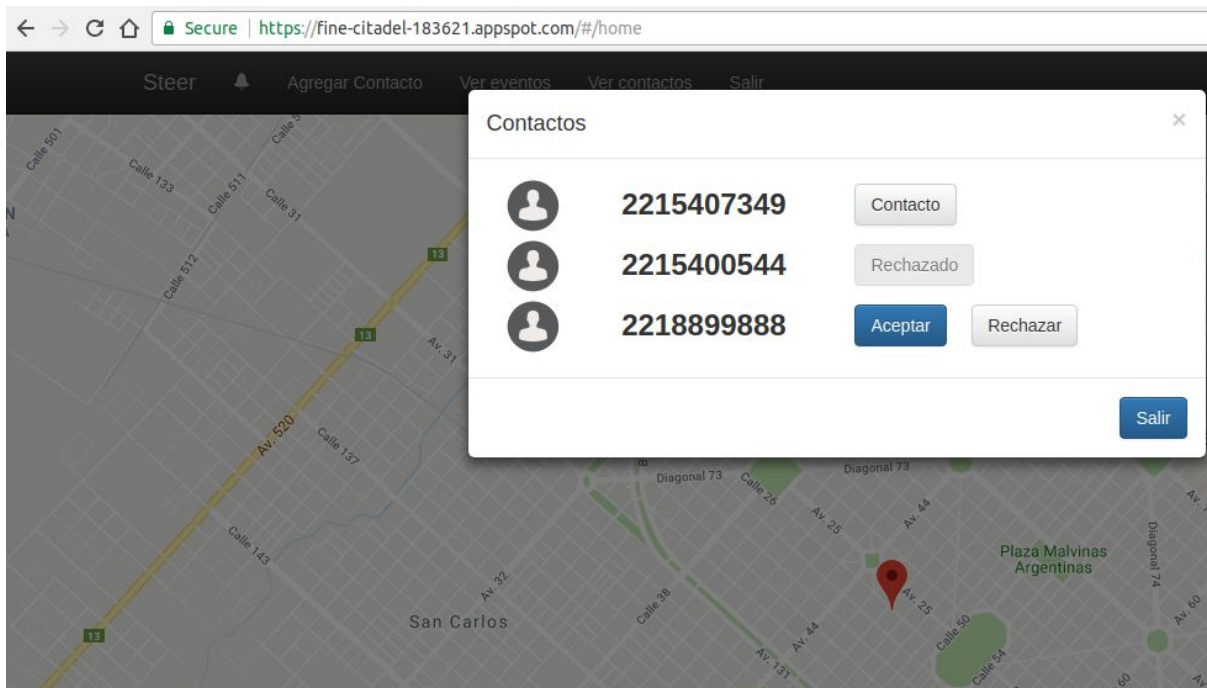


Figura 6.18. Menú contactos de sitio web Steer y diferentes estados.

En este menú se puede encontrar la información relacionada con los contactos del usuario. Consiste en una ventana modal que muestra en primera instancia la lista de contactos de la persona, este incluye una imagen de perfil y el nombre de usuario que puede ser o bien un apelativo o dirección de email del contacto. Al mismo tiempo la ventana modal nos muestra los usuarios que nos han enviado solicitud para establecer un vínculo de contacto, en estos casos se dispondrá de dos botones junto al nombre del usuario solicitante, que serán para aceptar dicha solicitud y agregarla a nuestra lista de contactos, o bien rechazar la misma y no mantener relación con este mismo. De esta manera este menú posee información tanto para consultar datos como para manipular una serie de ellos también.

- Salir

Este breve enlace cerrará la sesión de la aplicación, redirigiendo al usuario a la página de login.

La aplicación web se desarrolló íntegramente utilizando el IDE Eclipse, cuya plataforma nos fue muy útil para la estructura del proyecto, la instalación de plugins necesarios para funciones específicas y además para desplegar el mismo en el servicio de alojamiento web de Google utilizado. Es una plataforma para desarrollar aplicaciones que corren sobre la nube de Google, todo esto sobre un framework que permite solo concentrarnos en el código de la aplicación y no tener que lidiar con instalaciones y configuraciones, todo se hace de manera automática. Lo único que es necesario, es subir el código para tener la aplicación corriendo.

Al usar la infraestructura de Google es fácil de escalar sin importar los requerimientos que se necesiten (tráfico, almacenamiento, base de datos, usuarios, etc). Además nos provee una gran facilidad de integración con el entorno de desarrollo Eclipse, y el hecho que posea posibilidad de montar una aplicación de manera gratuita, su uso nos pareció una gran alternativa para el objetivo que teníamos planteado.

Para este sistema hemos decidido hacer una separación, una de ellas será el front-end, que es la parte con la que el usuario interactúa, y la otra es el back-end, encargada de procesar los datos del sistema. Esta separación nos ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas, y los transforma ajustándose a las especificaciones que demanda el back-end para poder procesarlos, devolviendo generalmente una respuesta que el front-end recibe y expone al usuario final de una forma entendible para este.

Por el lado del front-end se utilizó, html como lenguaje de marcado, estándar usado y reconocido por su uso masivo para el diseño de páginas web. Este efectúa una comunicación con AngularJS, framework que nos facilita la tarea de la comunicación entre el lenguaje Java que trata con los servicios REST y la información renderizada en la página Html. Por último se utilizó Bootstrap como framework para el diseño del sitio web, facilitandonos la tarea de hacer la aplicación responsive y todo tipo de herramientas para un diseño sencillo y fácil de manipular. Todas estas herramientas han sido descritas anteriormente en detalle, para tener un entendimiento general de que se está tratando y por qué el uso de las mismas.

Por otro lado, el back-end está realizado en Java como lenguaje de programación, para el manejo de todo el código que manipula la información del sistema. Con él logramos operar los pedidos que nos requiere el front-end, y además lo que concierna a persistencia de datos en la base de datos utilizada. Esta parte del sistema la vemos importante, ya que aquí se hacen procesamiento de datos a bajo nivel, que el usuario no debe enterarse que

existe, sino más bien, ver los resultados que éste arroja a la vista del sistema. La comunicación de datos se hace siempre en archivos con formato JSON, por una cuestión de practicidad en la manipulación de los mismos y, porque al estar familiarizado con éste se nos facilita la realización de operaciones.

Capítulo 7

Conclusiones y futuras líneas de investigación

7.1 Conclusión

En esta tesina hemos realizado un análisis de la problemática del uso de teléfonos móviles mientras se conduce y los peligros que esto conlleva. Observamos mediante estadísticas y datos de distintas organizaciones de tránsito que esto es un problema a nivel mundial.

En busca de soluciones que permitan reducir estos riesgos, se investigó qué tipo de tecnologías se podrían aplicar a esta problemática. Para esto se ha realizado una descripción del panorama actual en el campo de Ciudades inteligentes, y su relación con los vehículos inteligente y/o conectados.

Se planteó desarrollar una aplicación móvil que se comande directamente desde los dispositivos disponibles en vehículos modernos, que tienen controles como pantallas táctiles o mandos en el volante.

Se ha realizado un estudio del mercado de los distintos tipos de dispositivos IVI y cuales son las interfaces que ofrecen para lograr la comunicación con dispositivos móviles de manera tal de seleccionar cuál sería la opción más apropiada para el desarrollo que se propuso. En este punto se concluye que la mejor alternativa era una aplicación que se ejecute en el teléfono, cumpliendo la función de servidor, y sea transmitida y comandada desde el dispositivo IVI, funcionando como Cliente. Se analizaron los tres estándares de mayor penetración en el mercado automotriz determinando que la opción más adecuada era MirrorLink, principalmente porque es la opción más abierta y la única que no presentaba limitaciones en cuanto al desarrollo de la aplicación.

Ya en la etapa de diseño y desarrollo, se profundizó sobre las características del estándar MirrorLink y, como se pensó desde un principio, no se encontraron mayores inconvenientes para integrarlo al desarrollo, ya que se dispone de herramientas que automatizan todo el proceso. Lo más difícil, fue el proceso de pruebas en el simulador ofrecido por los desarrolladores de dicho estándar, que requería características obligatorias en cuanto al sistema operativo, versión e incluso una serie de kernel compatibles. En este punto, fue importante tener en cuenta las recomendaciones de diseño para la interfaz de

usuario, que sirven para minimizar las distracciones, y además son un requerimiento para certificar la aplicaciones.

Con Steer, logramos cumplir los objetivos planteados en la presente tesina. Se creó una aplicación que puede ser manipulada por los controles de los vehículos, evitando las distracciones de usar el teléfono mientras se conduce. Por otro lado, se integraron servicios ofrecidos por CeSPI como los sistemas SEM y CENIT logrando poner en valor y modernizar los mismos.

7.2 Futuras líneas de investigación

Luego de haber estudiado el área de ciudades inteligentes, más específicamente apuntando al software y los sistemas de infoentretenimiento de los automóviles, pudimos notar que ésta área en desarrollo de software, aún es muy reciente, ya que la cantidad de aplicaciones existentes, hablando a nivel mundial son muy pocas, dependiendo del estándar que tomemos varía esta cantidad, pero en general encontramos algunas pocas docenas en total, por lo cual entendemos que aún es un territorio muy reciente.

El caso de Steer surgió como una aplicación ideal para este tipo de tecnologías, ya que intenta aprovechar el automóvil como herramienta para facilitar el funcionamiento del software y, las variantes dentro del mismo que se puede obtener, siguiendo el lineamiento correcto de desarrollo.

Con respecto a trabajos futuros o futuras líneas de investigación, encontramos diferentes tipos de ideas que no estaban al alcance del presente proyecto, generalmente porque se utilizaban tecnologías que salían de la idea original de éste o bien porque las mismas ideas se alejaban del concepto original. Uno de los propósitos del trabajo se encuentra en la funcionalidad que el usuario denuncia de manera manual, un bacheo durante su conducción, la cual es eventualmente evaluada por un administrador y aceptada o rechazada según corresponda. Dicha funcionalidad, quisimos realizarla mediante el uso de sensores del dispositivo móvil, ya que el mismo contiene sensores de todo tipo, el mayor problema con esta solución es la imprecisión que tiene por ejemplo, el sensor de movimiento, que hubiese sido el más acertado para intentar de detectar un movimiento brusco en el automóvil, para así, lograr la detección automática de un bacheo. Por lo tanto, trabajar con el uso de un sistema OBD⁹ permitiría la obtención de datos precisos del automóvil, pudiendo mejorar algunos aspectos de la aplicación para hacerlo de manera automática, o bien agregando otro tipo de información que estos sensores puedan generar.

⁹ OBD (*On Board Diagnostics*) es un sistema de diagnóstico a bordo en vehículos, que aportan un monitoreo y control completo del motor y otros dispositivos del vehículo.

En cuanto a funcionalidades, la aplicación móvil puede añadir más opciones, hablando de denuncia de eventos que los que contempla el prototipo realizado. Siempre que pueda integrarse con el servicio de CENIT utilizado.

Referencias Bibliográficas

- [1] Vargas, Javier. (Enero 2012). *Bienvenido a la era del infoentretenimiento a bordo*. Auto Crash (15), p-52. CESVI Colombia. ISSN: 2145-8677. Accesible en:
<http://es.calameo.com/read/002609616ea1e2845262a>
- [2] *Uso del celular al volante: un problema creciente de distracción del conductor*. (2011). Organización Mundial de la Salud. Ginebra, Suiza. Accesible en:
http://www.who.int/violence_injury_prevention/publications/road_traffic/distracted_driving_public/es/
- [3] Harrison, C. y Donnelly, I.A. (2011) *A Theory of Smart Cities*. IBM Corporation. Accesible en:
<http://www.interindustria.hu/ekonyvtar/en/Smart%20cities%20and%20communities/Publications/A%20theory%20of%20smart%20cities.pdf>
- [4] Bayo, Jaime Gutiérrez. (2016). *Estudios de casos internacionales de ciudades inteligentes*. Banco Interamericano de Desarrollo. Santander, España. Accesible en:
<https://publications.iadb.org/bitstream/handle/11319/7717/Estudios-de-casos-internacionales-de-ciudades-inteligentes-Santander-Espana.pdf>
- [5] Flores, Dario Amar. (2016). *Estudios de casos internacionales de ciudades inteligentes*. Banco Interamericano de Desarrollo. Medellín, Colombia. Accesible en:
<https://publications.iadb.org/bitstream/handle/11319/7716/Estudios-de-casos-internacionales-de-ciudades-inteligentes-Medellin-Colombia.pdf>
- [6] *Smart Cities Readiness Guide - The planning manual for building tomorrow's cities today*. (2015). Smart Cities Council. Accesible en:
<https://smartcityalliance.ca/site/assets/files/1465/readiness-guide-v2-8-24-2015.pdf>
- [7] *Accelerating the next revolution in roadway safety*. (Septiembre 2016). National Highway Traffic Safety Administration (NHTSA). Accesible en:
http://www.nhtsa.gov/nhtsa/av/pdf/Federal_Automated_Vehicles_Policy.pdf

[8] *All Tesla Cars Being Produced Now Have Full Self-Driving Hardware*. (Octubre 2016). The Tesla Team. Accesible en:

https://www.tesla.com/blog/all-tesla-cars-being-produced-now-have-full-self-driving-hardware/?utm_campaign=GL_Blog_101916&utm_source=Twitter&utm_medium=social

[9] [10] Fraifer, Muftah y Fernström, Mikael. (2016). *Investigation of Smart Parking Systems and their technologies*. Accesible en:

<http://iot-smartcities.lero.ie/wp-content/uploads/2016/12/Investigation-of-Smart-Parking-Systems-and-their-technologies.pdf>

[11] Dave Evans (Abril 2011). *Internet de las cosas - Como la próxima evolución de Internet lo cambia todo*. CISCO. Accesible en:

https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf

[12] *GENIVI Compliant™ Products*. Accesible en: <https://www.genivi.org/compliant-products>

[13] Katarzyna Gajewska. (Octubre 2011). *Comarch joins the GENIVI Alliance*. Comarch. Accesible en:

<https://www.comarch.com/telecommunications/news/comarch-joins-the-genivi-alliance/>

[14] *Estudio Anual de coches conectados*. (2014). Interactive Advertising Bureau. Accesible en:

<http://www.iabspain.net/wp-content/uploads/downloads/2014/07/Informe-coches-conectados-2014.pdf>

[15] Bamiduro, W y Van der Meulen, R. (Febrero 2018). *Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017*. Egham, United Kingdom. Accesible en: <https://www.gartner.com/newsroom/id/3859963>

[16] Android Developers. (Enero 2018). *Paneles de control*. Accesible en:

<https://developer.android.com/about/dashboards/index.html#Platform>

[17] Apple Developer. *Human Interface Guidelines*. Accesible en:

<https://developer.apple.com/carplay/human-interface-guidelines/overview/introduction/>

- [18] Mirrorlink. (2016) . *Developers*. Accesible en: <https://mirrorlink.com/Developers>
- [19] Dr. Jörg Brakensiek (2013). *MirrorLink Technical Workshop*. Car Connectivity Consortium. Accesible en: <https://causeway.carconnectivity.org/wg/DevWG/document>
- [20] Google Maps API (2017). *Documentación, Guías, Servicios Web de las Google Maps API*. Accesible en: <https://developers.google.com/maps/web-services/overview?hl=es-419>
- [21] Firebase Realtime Database. (2018). *Documentación*. Accesible en: <https://firebase.google.com/docs/database/?hl=es-419>
- [22] Firebase Authentication. (2018). *Documentación*. Accesible en: <https://firebase.google.com/docs/auth/?hl=es-419>
- [23] Bootstrap. *Introducción*. Accesible en: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- [24] Eclipse. *The Eclipse Foundation open source community website*. Accesible en: <https://www.eclipse.org/>
- [25] Shyam Seshadri y Brad Green (2014). *AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps*. O'Reilly, ISBN: 978-1-491-90194-6.
- [26] Google Cloud. *Google App Engine*. <https://cloud.google.com/appengine/?hl=es>
- [27] Android Studio. (2018). *Conoce Android Studio*. Accesible en: <https://developer.android.com/studio/index.html>

Glosario

IVI - In-Vehicle Infotainment

CeSPI - Centro Superior para el Procesamiento de la Información

SEM - Sistema de Estacionamiento Medido

CENIT - Central de Información

SO - Sistema Operativo

GPS - Sistema de Posicionamiento Global

CO2 - Dióxido de carbono

TIC - Tecnologías de Información y Comunicación

SIMM - Sistema Inteligente de Movilidad de Medellín

SMS - mensajes cortos

CCTV - Circuito cerrado de televisión

RFID - Identificación por radiofrecuencia

IoT - Internet de las cosas

CD - Disco compacto

USB - Bus Universal en Serie

SDK - Kit de Desarrollo de Software

OAA - Open Automotive Alliance

NFC - Comunicación de campo cercano

iAP2 - iPod Accessory Protocol

HID - Dispositivo de interfaz humana

RTP - Protocolo de tiempo real

UPnP - Universal Plug and Play

VNC - Computación Virtual en Red

P2P - Red de pares

IPv4 - Cuarta versión del Protocolo de Internet

ARP - Protocolo de resolución de direcciones

TCP - Protocolo de control de transmisión

UDP - Protocolo Universal de Datos

DHCP - Protocolo de configuración dinámica de host

CCC - Car Connectivity Consortium

UI - Interfaz de Usuario

CVS - Sistema de Control de Versiones

MVC - Modelo Vista Controlador

IDE - Entorno de Desarrollo Integrado

API - Interfaz de Programación de Aplicaciones